

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

PCT/JP03/12065

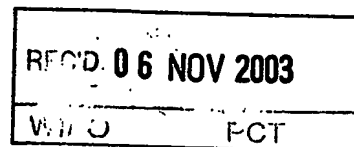
22.09.03

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日  
Date of Application: 2002年 9月20日

出 願 番 号  
Application Number: 特願2002-274539  
[ST. 10/C]: [JP2002-274539]



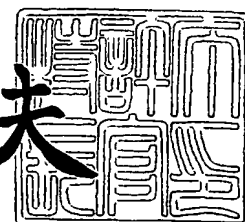
出 願 人  
Applicant(s): 松下電器産業株式会社

**PRIORITY  
DOCUMENT**  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH RULE 17.1(a) OR (b)

2003年10月23日

特許庁長官  
Commissioner,  
Japan Patent Office

今 井 康 夫



【書類名】 特許願

【整理番号】 2032740105

【提出日】 平成14年 9月20日

【あて先】 特許庁長官殿

【国際特許分類】 H04J 3/16

【発明者】

    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

    【氏名】 田中 朗宏

【発明者】

    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

    【氏名】 森 俊也

【発明者】

    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

    【氏名】 影本 英樹

【発明者】

    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

    【氏名】 山口 晃一郎

【発明者】

    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

    【氏名】 寺田 佳久

【特許出願人】

    【識別番号】 000005821

    【氏名又は名称】 松下電器産業株式会社

【代理人】

【識別番号】 100097445

【弁理士】

【氏名又は名称】 岩橋 文雄

【選任した代理人】

【識別番号】 100103355

【弁理士】

【氏名又は名称】 坂口 智康

【選任した代理人】

【識別番号】 100109667

【弁理士】

【氏名又は名称】 内藤 浩樹

【手数料の表示】

【予納台帳番号】 011305

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9809938

【書類名】 明細書

【発明の名称】 データ差替装置

【特許請求の範囲】

【請求項 1】 外部から繰り返し入力されるデータを取得する入力データ受信手段と、入力されたデータが差替対象であるかどうかを判別する差替元パケット判別手段と、差替後のデータを格納する差替後パケット保持手段と、入力データが差替対象であった場合には差替後パケット保持手段に格納されているデータに差替えて出力し、入力データが差替対象でない場合には入力されたデータを出力するモジュール差替手段とを具備し、外部から差替対象データが入力されると、差替後のデータに差替えて出力することを特徴とするデータ差替装置。

【請求項 2】 外部からの入力データに含まれる差替対象のデータサイズよりも、差替後データのサイズが小さい場合、入力される差替対象データと出力する差替後データの出力回数を等しくし、入力される差替対象データが出力する差替後データよりも大きい分は空のデータを出力することを特徴とする請求項 1 記載のデータ差替装置。

【請求項 3】 外部からの入力データが分割されて入力される場合、外部からの入力データに含まれる差替対象の分割データのサイズよりも、差替後データの分割データのサイズが以下の場合には、差替対象の分割データが入力された時に差替対象の分割データを出力し、外部からの入力データに含まれる差替対象の分割データのサイズよりも、差替後データの分割データのサイズが大きい場合には、空のデータを出力することを特徴とする、請求項 1 記載のデータ差替装置。

【請求項 4】 入力データを一時的に格納する入力パケット保持手段を保持し、外部からの入力データが分割されて入力され、分割データの単位でまとめて出力する必要がある場合、差替対象データの分割データが入力された時に、差替後データの分割データを入力された入力された差替対象データの分割データのサイズよりも多いか等しい分だけ出力し、モジュール差替手段は、以降の入力に差替対象の分割データが入力された時に、差替対象の分割データを出力せずに、入力パケット保持手段に一時的に格納されている入力データを出力することで、前記出力データを前記入力データよりも多く出力した差分を打ち消すことを特徴とす

る、請求項 1 記載のデータ差替装置。

【請求項 5】 差替えない入力データを入力される都度入力パケット保持手段に格納するのではなく、前記データを一度だけ取得し、差替後パケット保持手段に格納しておき、入力パケット保持手段に一時的に格納されている入力データを出力するかわりに、前記差替後パケット保持手段に格納したデータを出力することを特徴とする請求項 4 記載のデータ差替装置。

【請求項 6】 予め入力パケット保持手段に一定量のデータを格納しておき、通常時には入力データは入力パケット保持手段に格納され、前記一定量のデータが出力された後に出力され、差替対象の分割データが入力された時には、入力された差替対象の分割データ総量が差替後の分割データサイズ以上になった時に出力を開始し、前記差替後の分割データを出力するまでは、前記入力パケット保持手段に格納されている入力データを前倒しで出力することを特徴とする、請求項 4 記載のデータ差替装置。

【請求項 7】 予め入力パケット保持手段に一定量のデータを格納しておき、通常時には入力データは入力パケット保持手段に格納され、前記一定量のデータが出力された後に出力され、差替対象の分割データが入力された時には、入力された差替対象の分割データ総量が差替後の分割データサイズの半分になった時に出力を開始し、前記差替後の分割データを出力するまでは、前記入力パケット保持手段に格納されている入力データを前倒しで出力し、入力された差替対象の分割データ総量を越えて前記差替後の分割データを出力した後は、入力に差替対象以外のデータが入力されても差替対象の分割データが入力されたとカウントしないことを特徴とする、請求項 4 記載のデータ差替装置。

【請求項 8】 複数の差替対象データを、それぞれに 1 対 1 で対応する複数の差替後データに差替えることを特徴とする、請求項 4 記載のデータ差替装置。

【請求項 9】 差替対象データごとに独立して差替対象データの入力と差替後データの出力をカウントし、入出力のデータ量を等しくするように調整することを特徴とする、請求項 8 記載のデータ差替装置。

【請求項 10】 差替対象データに優先的に出力する差替対象データ 1 つを設定し、前記差替対象データが入力されると、優先的に差替後データを出力するこ

とを特徴とする、請求項 9 記載のデータ差替装置。

【請求項 11】 差替対象データ間に優先度を設定し、優先度の高い差替対象データほど、優先的に差替後データを出力することを特徴とする、請求項 9 記載のデータ差替装置。

【請求項 12】 差替えないデータにも差替対象データに対する優先度を設定し、優先度の高いデータほど、優先的に出力することを特徴とする、請求項 11 記載のデータ差替装置。

【請求項 13】 差替対象データ個々に加えて、差替対象データ全体のまとまりとして、差替対象データの入力と差替後データの出力をカウントし、入出力のデータ量を等しくするように調整することを特徴とする、請求項 8 記載のデータ差替装置。

【請求項 14】 外部から繰り返し入力されるデータを取得する入力データ受信ステップと、入力されたデータが差替対象であるかどうかを判別する差替元パケット判別ステップと、差替後のデータを格納する差替後パケット保持ステップと、入力データが差替対象であった場合には差替後パケット保持ステップで格納したデータに差替えて出力し、入力データが差替対象でない場合には入力されたデータを出力するモジュール差替ステップとから成り立つデータ差替方法。

【請求項 15】 外部から繰り返し入力されるデータを取得する入力データ受信ステップと、入力されたデータが差替対象であるかどうかを判別する差替元パケット判別ステップと、差替後のデータを格納する差替後パケット保持ステップと、入力データが差替対象であった場合には差替後パケット保持ステップで格納したデータに差替えて出力し、入力データが差替対象でない場合には入力されたデータを出力するモジュール差替ステップとが記録された記憶媒体。

【請求項 16】 外部から繰り返し入力されるデータを取得する入力データ受信プログラムと、入力されたデータが差替対象であるかどうかを判別する差替元パケット判別プログラムと、差替後のデータを格納する差替後パケット保持プログラムと、入力データが差替対象であった場合には差替後パケット保持プログラムで格納したデータに差替えて出力し、入力データが差替対象でない場合には入力されたデータを出力するモジュール差替プログラムとからなる、データ差替プ

ログラム。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、デジタルデータの伝送において、繰り返し入力されるデータを、ビットレートを変更せずに、入力されたデータ内容を差替えて出力するデータ差替装置に関する。

【0 0 0 2】

【従来の技術】

従来のテレビ放送の映像差替方法として、発局からのCM素材を受局のCM素材に差替えるときに、発局のCM素材の一部が一瞬見えること防止する方法として、各CM素材の始まりに前識別コードを、終わりに後識別コードを付加し、前識別コードと後識別コードの境界を検出し、その検出方法によって、受局の同期信号に同期するように遅延させた発局のCM素材を受局のCM素材に差替える方法がある（例えば、特許文献1参照）。

【0 0 0 3】

また、デジタルデータ入力を、内容をバッファリングする等して、入力されたデータを異なるデータに差替えて出力する技術が考えられている（例えば、特願 2 0 0 1 - 2 0 7 4 8 5 号参照）。

【0 0 0 4】

【特許文献1】

特開 2 0 0 1 - 4 5 3 7 1 号公報

【0 0 0 5】

【発明が解決しようとする課題】

近年、デジタル放送では、映像・音声以外の素材（データ放送素材）が放送されることが増えてきている。データ放送素材は、映像・音声素材と異なり、カルーセル伝送方式（MPEG-2規格 — ISO/IEC 13818-6 D S M - C C 参照）と呼ばれる伝送方式で送出される。カルーセル伝送方式は、同一のデータを繰り返し送出することで、受信側は送出されるデータの1周期分を

待てば、いつでも所望する情報を取得できることを特徴としている。

【0006】

一方、地上デジタル放送では、テレビ局毎に放送局系列があるため、中央のテレビ局（キー局）が放送したデータ放送内容を、地方のテレビ局（地方局）でも同時に放送する場合がある。この際に、キー局は、放送すると同時にデータ放送を地方局に送信する方法が考えられている。通常の運用では、キー局の番組放送時刻と地方局の番組放送時刻は同じであるため、地方局では、キー局から受信したデータ放送をリアルタイムに放送波で送出する必要がある。

【0007】

上記運用において、天気予報などのコンテンツに対して、キー局が関東地方の天気を放送している時に、地方局では関西地方の天気を放送する等の要求がある。この要求を満たすためには、キー局から送られてくるデータの一部（関東地方の天気）を、地方局で別のデータ（関西地方の天気）に差替えて送出することが要求される。

【0008】

データの一部を差替える際に、他のコンテンツの応答速度を保証するために、差替えないデータのレートは変更せず出力することが要求される場合がある。従来技術では、入力データと出力データのレートを変更せずに一部のデータを差替えると、差替え前のデータサイズよりも差替後のデータサイズが大きいと、差替えないデータの入力に対する出力タイミングが累積的に遅延してしまう。

【0009】

本発明は、入力レートと出力レートを一致させつつ、入力データの一部を別のデータに差替えて出力する装置を提供する。前記装置は、差替えないデータの遅延を最小限に抑えた出力や、優先度の高いデータの遅延を最小限に抑えた出力等、データを差替えたことによる出力のずれをコントロール可能であることを特徴とする。

【0010】

【課題を解決するための手段】

上記目的を達成するための本発明の装置は、データ差替装置であって、入力デ



ータ受信手段、差替元パケット判別手段、差替後パケット保持手段、モジュール差替手段、入力パケット保持手段から構成されることを特徴とする。

#### 【0011】

データ差替装置は、入力データ受信手段で繰り返し入力されるデータ（このデータはパケットの列として入力される）を取得し、取得したデータが差替える対象のパケット（差替対象パケット）であるかそうでないかを差替元パケット判別手段で判別する。モジュール差替手段は、差替元パケット判別手段によって入力パケットが差替対象パケットであると判断された場合そのパケットを破棄し、差替対象パケットではないと判断された場合入力されたパケットを入力パケット保持手段に格納する。前記モジュール差替手段は、更に、入力パケットが差替対象パケットかそうでないかと、それまでのパケット出力状況に応じて、差替後パケット保持手段、または、入力パケット保持手段に格納されているパケットを選択して出力する。

#### 【0012】

##### 【発明の実施の形態】

以下、本発明の実施の形態について図面を用いて詳細に説明する。

#### 【0013】

##### （実施の形態1）

図1は、本実施の形態に係るデータ差替装置10の構成を表すブロック図である。入力データ受信手段100は外部とのインタフェースを有し、外部からデータカルーセル伝送方式で入力されるTSパケットストリーム（ISO/IEC 13818-1 MPEG-2 system参照）を受信する。受信したパケットは、差替元パケット判別手段101、モジュール差替手段102へ渡される。

#### 【0014】

データカルーセル伝送方式においては、データは188オクテットのパケットを最小単位として伝送される。パケットが複数集まってセクションと呼ばれる単位を構成する。セクションは最低1パケットから最大23パケットから成り立つ。セクションが複数集まってデータとして意味のある最小単位であるモジュール

という単位を構成する。

#### 【0015】

セクションの先頭の packets は、そのセクションがどのモジュールを構成するセクションであるか、モジュール中何番目のセクションであるか、該セクションのサイズはいくつか等の情報を持つ。構成するモジュールを表す識別子を `moduleId`、モジュール中何番目のセクションであるかを表す識別子を `blockNumber`、セクションのサイズを表す識別子を `section_length` と呼ぶ。

#### 【0016】

図2は、入力データ受信手段100に、カルーセル伝送方式で入力されるデータの形式を図示したものである。一つの四角が一つの packets を表しており、左から順番に packets が伝送されてくる様子を表している。packets 200はPID=100、`moduleId`=0、`blockNumber`=0が付与された packets であることを表している。

#### 【0017】

また、`snum`=2から、PID=100で送出されている`moduleId`=0のモジュールは、2つのセクションから構成されることが分かり、`pnum`=3であることから、この packets を先頭とする3つの packets からセクションが構成されることが分かる。ただし、実際には、セクションの先頭の packets には、該セクションを構成する packets 数 (`pnum`) ではなくセクションの長さを表す`section_length`が記述されている。各 packets は固定長であるため、`section_length`を元にそのセクションを構成する packets 数は計算で一意に求められるため、以降の実施の形態では`section_length`の代わりに`pnum`を用いている。

#### 【0018】

次の packets 201は、PID=101、`moduleId`=0、`blockNumber`=0のセクション先頭の packets で、該モジュールは5つのセクションから構成されており (`snum`=5)、該セクションは4つの packets から構成されている (`pnum`=4) ことが記述されている。

## 【0019】

その次に送出されるパケット202、203には、PID=100であるという情報以外に、特にモジュールやセクションの情報は記述されていない。これは、直前のPID=100のパケット200にpnum=3が記述されているため、パケット200以降PID=100で送出されるパケットは、パケット200を含む3つのパケットが、moduleId=0、blockNumber=0のセクションを構成することが分かるため、PID以外の情報は不要であるためである。同様にパケット204、206、209もPID=101であるという情報のみが記述されている。

## 【0020】

パケット203で、PID=100のmoduleId=0、blockNumber=0のセクションの送出が完了するため、その次のPID=100のパケット205には、セクションの情報が記述されている必要がある。図2の例では、パケット205以降の4つのパケットで、moduleId=0、blockNumber=1のセクションが送出されていることが記述されている。

## 【0021】

以下同様に、セクションの先頭のパケットには、そのセクションの様々な情報と、該セクションを構成するパケット数が記述され、それ以外のパケットにはPIDのみの情報が記述されている。従って、あるセクションを送出し終える前に、同一PIDで別のセクションを送出開始してしまうと、pnumの値と実際に送出されるパケット数が異なり、受信時に矛盾を生じてしまう。このため、一旦セクションを送出し始めたら、そのセクションの最後のパケットを送出し終えるまでは、同一PIDで別のセクションを送出開始することはできない。

## 【0022】

図3にパケット、セクション、モジュールの関係を図示する。セクションは1個から23個のパケットで構成される。図3の例においては、パケット300～303の23個のパケットがPID=100で伝送されるblockNumber=0のセクション310を構成している。pnumの値が23であるため、セクション310が23個のパケットで構成されていることが分かる。

## 【0023】

モジュールは1個から255個のセクションで構成される。図3の例においては、モジュール320は、PID=100で伝送されるセクション310～312の45個のセクションから構成されている。snumの値が45であるため、モジュール320が45個のセクションから構成されていることが分かる。

## 【0024】

図4はデータ差替装置10への入力データと、データ差替装置10で差替えた後の出力データの例を表す図である。図4では、説明を単純にするために、入力データにはPID=100のパケットしか記述されていないが、他のPIDを持つパケットが入力されても構わない。入力データに他のPIDのパケットが入力されても、処理に本質的な違いはない。

## 【0025】

図4では、入力データのうちmoduleId=0のモジュールを別の内容のモジュールに差替える例を示している。入力データのうち400～402、407～409、413～415で示すパケットがmoduleId=0であり差替対象パケットである。差替後のモジュールは、差替後パケット保持手段103に予め格納されている。入力データが差替えようとしているモジュール（以後、差替対象モジュールと呼ぶ）、すなわち、moduleId=0のモジュールであった場合、それを差替後パケット保持手段103に格納されている差替後モジュールに差替えた出力が、出力データ41である。

## 【0026】

図5は差替後パケット保持手段103に格納されている差替モジュールを表す図である。差替対象モジュール50はblockNumber=0から9までの10個のセクションから構成されるモジュールであり、セクションごとに格納されている。

## 【0027】

図6は、データ差替装置10が、入力データを出力データに変換して出力する処理を表すフローチャートである。

## 【0028】

差替処理の開始時に、差替超過カウンタと呼ばれる値を 0 にセットする（ステップ S 6 0 0）。データ差替装置 1 0 は、差替超過カウンタの値を状態して持ち、差替超過カウンタと入力されたパケットを元に出力パケットを決定する。差替超過カウンタを 0 にセットしたら、入力パケットを取得し（ステップ S 6 0 1）、ステップ S 6 0 2 に遷移する。

#### 【0 0 2 9】

ステップ S 6 0 2 では、直前にどのようなパケットを出力したかを判定する。直前にセクションの最後以外にあたるパケットを出力していた場合にはステップ S 6 0 3 に遷移する。差替処理を開始した直後で直前に何もパケットを出力していない場合や、直前にセクションの最後にあたるパケットを出力していた場合には、ステップ S 6 1 0 に遷移する。

#### 【0 0 3 0】

ステップ S 6 0 3 では、直前に出力したパケットの次のパケットを出力し、ステップ S 6 0 4 に遷移する。ここで、直前に出力したパケットの次のパケットとは、直前に出力したパケット属するセクションを構成する次の構成パケットを指す。例えば、直前に出力したパケットが図 3 におけるパケット 3 0 0 のパケットならばパケット 3 0 1 を指し、直前に出力したパケットが図 3 におけるパケット 3 0 1 のパケットならばパケット 3 0 2 を指す。

#### 【0 0 3 1】

前記パケットは、直前に差替後パケット保持手段 1 0 3 に格納されているパケットを出力していた場合には、差替後パケット保持手段 1 0 3 に格納されており、直前に入力パケット保持手段 1 0 4 に格納されているパケットを出力していた場合には、入力パケット保持手段 1 0 4 に格納されており、直前に入力されたパケットを出力していた場合には、次のパケットも入力データから入力される。

#### 【0 0 3 2】

ステップ S 6 0 4 において、差替元パケット判別手段 1 0 1 で入力パケットが差替対象モジュールを構成するパケットであるか、そうでないかを判断し、入力パケットが差替対象モジュールを構成するパケットであった場合、ステップ S 6 0 5 に進み、入力パケットが差替対象モジュールを構成するパケットでなかった

場合、ステップS607に遷移する。

【0033】

ここで、差替対象モジュールとは、入力されるモジュールのうち、差替える対象となっているモジュールを指す。すなわち、`moduleId=0`のモジュールが差替対象であるならば、入力が`moduleId=0`のモジュールを構成するセクションを構成するパケットであった場合に真となる。

【0034】

ステップS605では、ステップS603において出力したパケットが、入力パケット保持手段104に格納されていたパケットであったかどうかを判断する。ステップS603において出力したパケットが入力パケット保持手段104に格納されていたパケットであった場合には、差替超過カウントから1を引き（ステップS606）、ステップS601に遷移する。ステップS603において出力したパケットが入力パケット保持手段104に格納されていたパケットでなかった場合には、何もせずステップS601に遷移する。

【0035】

ステップS607では、ステップS603において出力したパケットが、差替後パケット保持手段103に格納されていたパケットであったかどうかを判断する。ステップS603において出力したパケットが差替後パケット保持手段103に格納されていたパケットであった場合には、入力パケットを入力パケット保持手段104に格納し（ステップS608）、差替超過カウントに1を対して（ステップS609）ステップS601に遷移する。ステップS603において出力したパケットが差替後パケット保持手段103に格納されていたパケットでなかった場合には、何もせずステップS601に遷移する。

【0036】

ステップS602において、直前にセクション途中のパケットを出力していなかった場合や差替処理の開始時には、ステップS610へ進み、ステップS610において、差替元パケット判別手段101で入力パケットが差替対象モジュールを構成するパケットであるか、そうでないかを判断する。入力パケットが差替対象モジュールを構成するパケットであった場合、ステップS611に進み、入

力パケットが差替対象モジュールを構成するパケットでなかった場合、ステップ S 6 1 3 に遷移する。

#### 【0037】

ステップ S 6 1 1 において、差替超過カウン트가 0 ならば、ステップ S 6 1 2 に遷移し、差替後パケット保持手段 1 0 3 から次に送出するセクションの先頭パケットを出力し、ステップ S 6 0 1 に遷移する。ここで、次に送出するセクションとは、直前に差替後パケット保持手段 1 0 3 から出力したセクションの次のセクションを指す。差替処理開始時でまだ何もセクションを送出していなかった場合や直前にモジュールを構成する最後のセクションを送出していた場合には、差替後パケット保持手段 1 0 3 に格納されているモジュールの先頭のセクションを指す。例えば、直前に図 3 におけるセクション 3 1 0 を出力していた場合セクション 3 1 1 を指し、差替処理開始時でまだ何もセクションを送出していなかった場合や、直前に図 3 におけるセクション 3 1 2 を出力していた場合には、セクション 3 1 0 を指す。

#### 【0038】

ステップ S 6 1 1 において、差替超過カウン트가 0 を越えている場合、ステップ S 6 1 4 に進み、入力パケット保持手段 1 0 4 に格納されているパケットを出力する。入力パケット保持手段 1 0 4 は F I F O 型の領域であり、入力パケット保持手段 1 0 4 に格納されているパケットを出力する際には、格納された順番にパケットが出力される。ステップ S 6 1 4 でパケットを出力した後は、ステップ S 6 0 6 へ遷移する。

#### 【0039】

ステップ S 6 1 0 において、入力パケットが差替対象モジュールを構成するパケットでなかった場合、ステップ S 6 1 3 に進み、入力パケットを入力パケット保持手段 1 0 4 に格納し差替超過カウン트에 1 を加えた後（ステップ S 6 1 3）、入力パケット保持手段 1 0 4 に格納されているパケットを出力し（ステップ S 6 1 4）、ステップ S 6 0 6 に遷移する。

#### 【0040】

ステップ S 6 0 6 では、差替超過カウンタから 1 を引いて、ステップ S 6 0 1

に遷移する。

#### 【0041】

以下に、図4に示す入力データ40がデータ差替装置10に入力された場合に、`module_id=0`のモジュールを、図5に示す差替対象モジュール50に差替えて、出力データ41を出力する処理を例に説明する。

#### 【0042】

差替処理を開始すると、まずステップS600において、差替超過カウントを0にリセットし、最初の入力パケットとして、図4のパケット400を取得する(ステップS601)。直前に出力したパケットは存在しないため、ステップS602の判定で`no`となり、ステップS610へ遷移する。パケット400は、`module_id=0`なので、これは差替対象モジュールであるためステップS611へ遷移する。ステップS611においては、この時点での差替超過カウントは0であるため`no`となりステップS612へ遷移する。

#### 【0043】

それ以前にモジュールを差替えて出力していないため、差替後パケット保持手段103に格納されているセクションのうち、次に送出するセクションは、モジュールを構成する最初のセクション、すなわち、図5における`blockNumber=0`のセクションである。従って、ステップS612では、図5における`blockNumber=0`のセクションの先頭パケットであるパケット500を出力する。なお、前記パケット500は図4においては、パケット451にあたる。ステップS612においてパケットを出力した後、ステップS601へ遷移する。

#### 【0044】

次にステップS601で、入力データ受信手段100は図4におけるパケット401を取得する。直前には、パケット500を出力しており、このパケットは`blockNumber=0`のセクションを構成するパケットであり、このパケットで前記セクションが完了しているわけではないため、ステップS602での判定は`yes`となり、ステップS603に遷移する。

#### 【0045】



ステップS603では、直前に出力したパケット（図5におけるパケット500）の次のパケット、すなわち、図5におけるパケット501を出力し、ステップS604に遷移する。このパケットは、図4ではパケット451にあたる。

#### 【0046】

ステップS604において、差替元パケット判別手段101は、入力パケット（図4におけるパケット401）が差替対象モジュールを構成するパケットであるかどうかを判別する。パケット400が $p m u n = 3$ のセクションの1パケット目であるため、それ以降 $P I D = 100$ で入力される2つのパケットは、パケット400と同じセクションを構成するパケットであることが分かる。このことより、入力されたパケット401もパケット400と同様 $m o d u l e I d = 0$ のセクションを構成するパケットであると分かるため、ステップS604の判別結果は $y e s$ となり、ステップS605に遷移する。

#### 【0047】

この時に出力したパケットは、パケット501であり、それは入力パケット保持手段104に保持されていたパケットではなく、差替後パケット保持手段103に格納されているパケットであるため、ステップS605での判定は $n o$ となり、ステップS601へ遷移する。その次に、パケット402を受信した時もパケット401を受信した時と同様の処理を実施する。

#### 【0048】

次に、パケット403を受信した時に、ステップS604の判定で入力パケット403は $m o d u l e I d = 1$ を構成するパケットであり、差替対象モジュールを構成するパケットではないため、ステップS607に遷移する。また、この処理に至る過程で、ステップS603において送出したパケットは図5のパケット503であるため、差替後パケット保持手段103に格納されていたパケットである。

#### 【0049】

従って、ステップS607での判定は $y e s$ となり、ステップS608に遷移し、入力パケット（図4におけるパケット403）を入力パケット保持手段104に格納し（ステップS608）、差替超過カウントに1を加え（ステップS6

09)、ステップS601へ遷移する。この時点で、入力パケット保持手段104には、パケット403が一つ格納されている状態となる。

#### 【0050】

次に、パケット404、パケット405を受信した時にも同様の処理を行い、出力パケットとしてパケット504、パケット505を出力する。前記2つのパケット出力処理を実行した後は、差替超過カウンタは3になり、入力パケット保持手段104にはパケット403、パケット404、パケット405の3つのパケットが格納されている状態となる。

#### 【0051】

次に、ステップS601にて図4におけるパケット406を取得し、ステップS602に遷移する。直前に出力したパケットは、図5におけるパケット505であり、これはセクションの最後のパケットなので、ステップS602での判定はnoとなりステップS610に遷移する。

#### 【0052】

入力パケットは、図4におけるパケット406であるので、差替対象モジュールではない。従って、ステップS610での判定はnoとなり、ステップS613に遷移する。ステップS613では、入力パケットを入力パケット保持手段104に格納し差替超過カウンタに1を追加する。この処理によって、入力パケット保持手段104には、パケット403～パケット406の4つのパケットが格納され、差替超過カウンタは4という状態になる。

#### 【0053】

ステップS613の処理完了後、ステップS614に遷移し、入力パケット保持手段104からパケットを出力する。入力パケット保持手段104は、FIFOのメモリ領域であるため、最初に入力されたパケット403を出力して、ステップS606へ遷移し、差替超過カウンタから1を引いて、ステップS601へ遷移する。この時点での、差替超過カウンタは3で、差替後パケット保持手段104には、パケット404、パケット405、パケット406の3つのパケットが格納された状態になる。

#### 【0054】

次に図4におけるパケット407が入力されると、ステップS601でパケットを取得し、ステップS602で、直前に出力したパケットがパケット403であり、セクションの最終パケットではないためnoと判定されステップS610へ遷移する。

#### 【0055】

入力パケットは、パケット407でmoduleId=0のパケットであるため、ステップS610における判定はyesとなり、ステップS611へ遷移する。この時点での差替超過カウントは3であるため、ステップS611の判定はyesとなるためステップS614へ遷移する。その後、入力パケット保持手段104から、その時点で格納されているパケット列の先頭パケットであるパケット404を出力して（ステップS614）、差替超過カウントから1を引いて（ステップS606）、ステップS601へ遷移する。

#### 【0056】

この時点では、差替超過カウントは2で、差替後パケット保持手段103に格納されているパケットは、パケット405、パケット406の2つのパケットという状態になる。

#### 【0057】

その後、図4におけるパケット408、パケット409が入力された時には、上記と同様の処理を実行し、パケット409の処理を実行した後は、差替超過カウントが0で、差替後パケット保持手段103には何も格納されていない状態になる。

#### 【0058】

次にステップS601で、図4におけるパケット410を取得すると、直前に送出したパケットは、図4におけるパケット406（出力ではパケット459）なので、セクション最後のパケットであるため、ステップS602における判定はnoとなり、ステップS610に遷移する。

#### 【0059】

ステップS610では、入力パケット（図4におけるパケット410）は、moduleId=1のモジュールを構成するパケットであるため判定はnoとな

り、ステップS613に遷移する。その後、ステップS613、ステップS614、ステップS606と順に処理を行い、入力された packets をそのまま出力して、packet S601に遷移する。

#### 【0060】

次にステップS601で、図4におけるpacket 411を取得すると、直前に送出したpacketは、図4におけるpacket 410（出力ではpacket 460）なので、セクション最後のpacketでないため、ステップS602における判定はyesとなり、ステップS603に遷移する。

#### 【0061】

ステップS603においては、直前に出力したpacketがpacket 410であり、その次のpacketはpacket 411である。packet 411は、入力されたpacketであるので、入力されたpacketを単にそのまま出力し、ステップS604に遷移する。

#### 【0062】

入力されたpacket 411は、差替対象モジュールを構成するpacketではなく、module Id=1のモジュールを構成するpacketであるため、ステップS604での判定はnoとなり、ステップS607へ遷移する。ステップS603における処理は、入力packetをそのまま出力する処理であったため、ステップS607での判定はnoとなり、ステップS601へ遷移する。次に図4におけるpacket 412を取得した時の処理も、上記と同様である。

#### 【0063】

その次に、packet 413を受信した時の処理は、packet 400を受信した時の処理と同様で、packet 414、packet 415を受信した時の処理は、packet 401、packet 402を受信した時の処理と同様である。

#### 【0064】

このようにして、図6に示す処理フローチャートを用いて、図4における入力データ40が入力された時に、出力データ41を生成して出力することが可能である。

#### 【0065】

なお、入力データの `module Id` を変更して差替出力することも可能である。例えば、前記処理と同等の処理で、入力における `module Id=0` のモジュールを図 7 に示すモジュールに差替えることも可能である。その場合、入力は `module Id=0` のモジュールだが、出力は `module Id=3` のモジュールに差替えられる。

#### 【0066】

なお、差替えるモジュールの `module Id` を予め登録しておくのではなく、差替の設定に応じてリアルタイムに書き替えて出力することも可能である。例えば、入力における `module Id=0` のモジュールを図 5 に示すモジュールに差替えるが、パケットを出力する際に、`module Id` の値を 3 に書き替えて出力すれば、入力は `module Id=0` のモジュールだが、出力は `module Id=3` のモジュールに差替えられる。MPEG2 規格では、セクションの最後に CRC を付与する必要があるため、`module Id` の値を変更したら、CRC の値も変更する必要がある。前記 CRC の計算は、各パケットの内容が分かっているならば、容易に計算可能な値であるため、本明細では記述を省略する。

#### 【0067】

##### (実施の形態 2)

本実施の形態では、MPEG2 規格 (ISO/IEC 13818-6) におけるモジュール構成の特徴を利用することですること、より簡略化した装置で、入力されたモジュールよりもサイズの小さなモジュールへの差替を実現する方法を示す。

#### 【0068】

以下に、MPEG2 規格におけるモジュール構成の特徴を示す。MPEG2 規格では、モジュールは `block` (セクション) に分割して伝送されるが、セクションのサイズには制限がある。モジュールを構成する最後のセクションを除いた、残り全てのセクションのサイズは、指定された値 (`block Size`) に等しくなくてはならないという制限である。より厳密には、セクションのヘッダと CRC を除いたサイズが前記 `block Size` に等しくなるという制限であり、セクションのヘッダと CRC の合計は 30 オクテットである。ただし、セク

ションのヘッダとCRCに関しては、本発明と本質的な関係が無いため、以降の記述では、セクションのヘッダとCRCは無視して記述する。

#### 【0069】

例えば、blockSize=700であるとする、モジュールを構成するセクションは、最終セクションを除いて、全てサイズが700オクテットとなる。サイズが3000オクテットのモジュールは、700オクテットのセクションが4つと、200オクテットのセクション1つに分割されて伝送される。

#### 【0070】

さらに、MPEG2規格では、パケットのサイズは188オクテットでヘッダサイズが4オクテットであるため、1パケットあたり184オクテットのデータを伝送することが可能である。より厳密には、セクションの先頭を伝送するパケットには、セクション先頭を伝送していることを示すための情報が1オクテット分必要であるため、セクション先頭を伝送するパケットのみ183オクテットのデータが伝送できる。ただし、前記事項も本発明とは直接関係ないため、以降の記述では、セクション先頭を含むパケットでも184オクテットを伝送できるとする。

#### 【0071】

従って、700オクテットのセクションは4つのパケットから構成され、200オクテットのセクションは2つのパケットから構成される。図9は、blockSize=700の場合の、3000オクテットのモジュール（モジュール90）のセクション、パケット構成である。

#### 【0072】

モジュール90は、5つのセクションから構成される。各セクションには、blockNumberが0から順番に指定されており、最終セクション（セクション95）を除く4つのセクション（セクション91、セクション92、セクション93、セクション94）は、全て700オクテットであり、4つのパケットから構成されている。セクション95のみblockSizeよりも小さくなることが許されており、そのサイズは200オクテットである。1パケットで伝送できるデータサイズは184オクテットであるため、セクション95は2つのパ

ケットから構成されている。

#### 【0073】

図8は、本実施の形態に係るデータ差替装置80の構成を表すブロック図である。データ差替装置80は、データ差替装置10と比較して、入力 packets 保持手段を持たない点で異なる。以下に、本実施の形態で、MPEG2規格に従った入力データを差替える例を詳細に説明する。

#### 【0074】

図10は、入力データに含まれている3000オクテットのモジュール（図9に示すモジュールと同等の構成をしている）を2000オクテットのモジュール（図11に示すモジュールと同等の構成をしている）に差替える場合の、入出力の例である。本実施の形態では、入力されるモジュールのサイズが差替えて出力するモジュールのサイズよりも大きい場合にのみ適用可能である。

#### 【0075】

本実施の形態は、図10に示すように、差替対象モジュールが1つ入力されるごとに差替後のモジュールを1つ出力する。入力されるモジュールの方が差替えて出力するモジュールよりもサイズが大きいため、余剰部分はNULLパケットと呼ばれるダミーのパケットに置き換えて送出する。本実施の形態を実施することによって、入力モジュール1回につき出力モジュールを1回出力することができる。

#### 【0076】

図12は、本実施形態におけるデータ差替装置80の処理を表すフローチャートである。データ差替装置80は、差替処理を開始すると、ステップS1200において、入力データ受信手段800が入力されたパケットを取得し、ステップS1201に遷移する。ステップS1201において、差替元パケット判別手段801が、入力パケットが差替対象モジュールを構成するパケットであるかどうかを判定する。入力パケットが差替対象モジュールを構成するパケットであった場合には、ステップS1202へ遷移し、そうでない場合には、ステップS1206へ遷移する。

#### 【0077】

ステップS1202において、モジュール差替手段802は、 $in\_bnum < out\_snum$ であるかどうかを判定する。ここで、 $in\_bnum$ とは、入力パケットが構成するセクションの**blockNumber**を表し、 $out\_snum$ とは、差替後のモジュールを構成するセクション数を表すものとする。前記条件を満たす場合は、ステップS1203へ遷移し、そうでない場合ステップS1205へ遷移する。

#### 【0078】

$in\_bnum < out\_snum$ である場合、ステップS1203で  $in\_pos \leq out\_pnum$  であるかどうかをモジュール差替手段802で判定する。ここで、 $in\_ppos$ とは、入力されたパケットが構成するセクションの何番目のパケットであるかを表し、 $out\_pnum$ とは、差替えるモジュールにおいて、前記  $in\_bnum$  番目のセクションに対応するセクションを構成するパケットの数を表すものとする。

#### 【0079】

ステップS1203で **yes** と判定された場合、ステップS1204へ遷移し、入力されたパケットに差替えて、差替後パケット保持手段803に格納されている差替えるモジュールのうち、**blockNumber**が  $in\_bnum$  に等しいセクションの、 $in\_ppos$  の位置に相当するパケットを出力し、ステップS1200へ遷移する。

#### 【0080】

ステップS1202、ステップS1203で **no** と判定された場合、ステップS1205へ遷移し、入力されたパケットに差替えて、**NULL**パケットを出力し、ステップS1200へ遷移する。

#### 【0081】

以下に、図10に示す入力データ1000がデータ差替装置80に入力された場合に、 $module\_id=0$  のモジュールを、図11に示す差替対象モジュール1100に差替えて、出力データ1050を出力する処理を例に説明する。

#### 【0082】

データ差替装置80は、差替処理開始後に、図10におけるパケット1001



が入力されると、ステップS1200において入力データ受信手段800がパケット1001を取得する。取得したパケット1001は、差替元パケット判別手段801によって差替対象のパケットであると判定され（ステップS1201）、ステップS1202へ遷移する。

#### 【0083】

パケット1001の場合、 $in\_bnum = 0$ 、 $out\_snum = 3$ であり、 $in\_bnum < out\_snum$ であるため、ステップS1202でyesと判定され、ステップS1203に遷移する。

#### 【0084】

ステップS1203において、 $in\_ppos \leq out\_pnum$ であるためyesと判定され、ステップS1204へ遷移する。パケット1001はセクションの先頭パケットであるため $in\_ppos = 0$ となる。また、パケット1001はblockNumber=0のセクションを構成するパケットであるため、差替えるモジュール（図11で示すモジュール1100）のblockNumber=0のセクションを構成するパケット数が $out\_pnum$ となる。従って、セクション1110を構成するパケット数=4が $out\_pnum$ の値となる。

#### 【0085】

次に、ステップS1204において、モジュール差替手段802が差替後パケット保持手段803からパケット1111を読み出して出力する。これは、入力パケットがblockNumber=0のセクションの先頭パケットであるため、それに対応するパケットが、差替後パケット保持手段803に格納されているモジュール中の、blockNumber=0のセクション（セクション1110）の先頭パケットであるパケット1111であるためである。ステップS1204でパケットを出力した後は、ステップS1200へ遷移する。

#### 【0086】

図10におけるパケット1002、パケット1003、パケット1004が入力された時の処理は、パケット1001が入力された時と同様の処理を実施する。

## 【0087】

その後、パケット1005が入力されると、ステップS1200で入力データ受信手段800がパケット1005を取得し、ステップS1201に遷移する。パケット1005は`moduleId=1`であるため、ステップS1201において差替元パケット判別手段801は`no`と判定し、ステップS1206へ遷移する。その後、ステップS1206において、入力されたパケットをそのまま出力しステップS1200へ遷移する。

## 【0088】

次に入力されるパケット1009からパケット1015に対しては、パケット1001が入力された時と同様の処理を実施する。

## 【0089】

次にパケット1016が入力されると、ステップS1200において、入力データ受信手段800がパケット1016を取得し、ステップS1201で差替元パケット判別手段801が該パケットは差替対象であると判定し、ステップS1202へ遷移する。パケット1016は、`in_bnum=2`、`out_snum=3`であるため、ステップS1202での判定は`yes`となり、ステップS1203へ遷移する。

## 【0090】

パケット1016は、`in_ppos=4`で`out_pnum=3`であるため、ステップS1203における判定は`no`となり、ステップS1205へ遷移する。ステップS1205において、モジュール差替手段802は、入力されたパケットを廃棄し、NULLパケットを出力する。前記処理終了後、ステップS1200へ遷移する。

## 【0091】

パケット1017の入力に対しては、パケット1005と同様の処理を実施し、入力されたパケットをそのまま出力する。

## 【0092】

次にパケット1018が入力されると、入力データ受信手段800でパケット1018を取得する。取得したパケットは、差替対象であるため、ステップS1

201でyesと判定し、ステップS1202へ遷移する。パケット1018のin\_bnumは3であり、out\_snum=3であるため、ステップS1202の判定はnoとなり、ステップS1205へ遷移し、モジュール差替手段802はNULLパケットを出力する。前記処理終了後、ステップS1200へ遷移する。

#### 【0093】

次のパケット1019からパケット1023が入力されると、パケット1018と同様の処理を実施し、パケット1024が入力されると、パケット1005と同様の処理を実施する。

#### 【0094】

上記処理で、図10における入力データ1000が入力された時に、図12に示す処理フローチャートに基づいて、データ差替装置80で出力データ1050を出力することが可能である。

#### 【0095】

以上に示すように、本実施形態によれば、差替対象モジュールのサイズが、差替後のモジュールサイズよりも大きな場合に、差替えないモジュールは入力と全く同様に出力し、かつ、入力される差替対象モジュールの送出回数を変えずに別のモジュールに差替えることが可能となる。

#### 【0096】

なお、本実施形態は、入力データがMPEG2規格に従っている必要は無く、MPEG2規格におけるblock sizeにあたる制限があり、データを分割して繰り返し送出する伝送方式で入力されていれば、適用可能であることは言うまでもない。

#### 【0097】

なお、MPEG2規格以外の伝送方式で入力されるデータが、パケット単位で分割されておらず、セクション単位でまとまって伝送されてくる場合でも、本実施の形態が有効である。

#### 【0098】

(実施の形態3)

実施の形態2では、モジュールサイズが小さくなった分は、NULLパケットに差替えて送出するため、モジュールの送出回数は入力に従うことが可能であるが、NULLパケットに差替える分、入力に対して有効帯域が減少してしまう。また、差替えるモジュールサイズが入力される差替対象モジュールよりも大きな場合には、単純に実施の形態2を適用することはできない。

#### 【0099】

本実施の形態は、出力するパケットを選択する処理においてのみ、実施の形態2と異なる。以下に、実施の形態2と異なる部分についてのみ説明する。

#### 【0100】

図13は、本実施形態におけるデータ差替装置80の処理を表すフローチャートである。データ差替装置80は、差替処理を開始すると、ステップS1300において、差替後パケット保持手段803に格納されている差替対象モジュールの、blockNumber=0のセクションを出力対象セクションと設定し、ステップS1301へ遷移する。

#### 【0101】

ステップS1301では、入力データ受信手段800が入力されたパケットを取得し、ステップS1302に遷移する。ステップS1302において、差替元パケット判別手段801が、入力パケットがセクションの先頭のパケットであるかどうかを判定する。入力パケットがセクションの先頭のパケットである場合、ステップS1303へ遷移し、そうでない場合には、ステップS1308へ遷移する。

#### 【0102】

ステップS1303では、差替元パケット判別手段801が、入力パケットが差替対象モジュールを構成するセクションの先頭パケットであるかどうかを判定する。入力パケットが差替対象モジュールを構成するセクションの先頭パケットでない場合、ステップS1304へ遷移する。

#### 【0103】

ステップS1304では、モジュール差替手段802が、まずステップS1303で差替対象モジュールを構成するセクションの先頭パケットでないと判定さ

れたパケットを、そのまま出力する。次に、入力データ受信手段 800 が入力されるパケットを取得し、差替元パケット判別手段 801 が、前記パケットがセクション先頭パケットであるかどうかを判定する。

#### 【0104】

前記パケットがセクション先頭パケットでない場合、前記パケットをそのまま出力し、次のパケット受信処理を行い、セクションの先頭パケットを受信するまで、受信パケットをそのまま出力する処理を繰り返す。セクションの先頭パケットを受信すると、ステップ S1303 へ遷移する。

#### 【0105】

ステップ S1303 において、差替元パケット判別手段 801 が、受信したパケットは差替対象モジュールを構成するセクションの先頭パケットであると判定した場合、ステップ S1305 へ遷移する。ステップ S1305 において、モジュール差替手段 802 は、 $in\_pnum \geq out\_pnum$  であるかどうかを判定する。ここで、 $in\_pnum$  は入力されたパケットを含むセクションを構成するパケットの数で、 $out\_pnum$  は出力対象セクションを構成するパケットの数を示すものとする。ステップ S1305 で  $no$  と判定した場合ステップ S1308 へ遷移する。

#### 【0106】

ステップ S1305 で  $yes$  と判定した場合、ステップ S1306 へ遷移する。ステップ S1306 においては、モジュール差替手段 802 が入力されたパケットを、差替後パケット保持手段 803 に格納されている出力対象セクションを構成するパケットに、順番に差替えて出力する。ステップ S1306 の処理は、出力対象セクションを全て出力し終わるまで繰り返し実行する。ステップ S1306 の処理が終了すると、ステップ S1307 へ遷移し、ステップ S1306 で出力したセクションの次のセクションを、新たな出力対象セクションと設定する。ただし、ステップ S1306 において出力したセクションが、差替対象モジュールを構成する最終セクションであった場合、差替対象モジュールを構成するセクションのうち、先頭のセクションを、新たな出力対象セクションとする。

#### 【0107】

ステップS1302においてnoと判定された場合、ステップS1305においてnoと判定された場合、およびステップS1307の処理実行後には、ステップS1308へ遷移する。ステップS1308においては、まず入力データ受信手段800が入力されるパケットを取得する。取得したパケットが、セクションの先頭パケットで無い場合、モジュール差替手段802が、そのパケットをNULLパケットに差替えて出力する。前記パケット取得処理とNULLパケットに差替えて出力する処理は、セクション先頭のパケットが入力されるまで繰り返す。セクション先頭のパケットが入力されると、ステップS1308の処理を終了して、ステップS1303へ遷移する。

#### 【0108】

以下に、図14に示す入力データ1400がデータ差替装置80に入力された場合に、module\_\_id=0のモジュールを、図11に示す差替対象モジュール1100に差替えて、出力データ1450を出力する処理を例に説明する。

#### 【0109】

データ差替装置80は、差替処理開始後、ステップS1300において差替後のモジュール（差替後パケット保持手段803に格納されているモジュール）のblockNumber=0のセクション（セクション1110）を出力対象セクションとし、ステップS1301へ遷移する。

#### 【0110】

ステップS1301において、図14におけるパケット1401が入力されると、入力データ受信手段800がパケット1401を取得する。その後、差替元パケット判別手段801は、取得したパケット1401を、セクションの先頭パケットであると判定し（ステップS1302）ステップS1303へ遷移する。ステップS1303において、差替元パケット判別手段801は、入力パケット1401が差替対象モジュールを構成するセクションの先頭パケットであると判定し（ステップS1303）、ステップS1305へ遷移する。

#### 【0111】

パケット1401は4つのパケットから構成されるセクションの先頭パケットであるためin\_\_pnum=4であり、出力対象セクションのセクション111

1は4つのパケットから構成されているためout\_pnum=4となる。in\_pnum>=out\_pnumの条件を満たすため、ステップS1305ではyesと判定し、ステップS1306へ遷移する。

#### 【0112】

ステップS1306では、モジュール差替手段802は、出力対象セクションであるセクション1110を送出し終えるまで、入力パケットを出力対象セクションのパケットに差替えて出力する。この場合、入力されるパケット1401、パケット1402、パケット1403、パケット1404を、それぞれ、差替後パケット保持手段803に格納されているパケット1111、パケット1112、パケット1113、パケット1114に差替えて出力する。前記パケットは、図14における、パケット1451からパケット1454にあたる。

#### 【0113】

ステップS1306の処理を完了するとステップS1307へ遷移し、送出したセクションの次のセクションである、セクション1112を出力対象セクションとし、ステップS1308へ遷移する。ステップS1308において、入力データ受信手段800が次の入力パケットであるパケット1405を取得する。パケット1405はセクション先頭のパケットであるため、ステップS1303へ遷移する。

#### 【0114】

ステップS1303において、入力パケット1405は差替対象モジュールを構成するセクションの先頭パケットではないため、差替元パケット判別手段801がnoと判定し、ステップS1304へ遷移する。

#### 【0115】

ステップS1304では、次にセクション先頭のパケットが入力されるまで、モジュール差替手段802が入力されたパケットをそのまま出力する。従って、パケット1405からパケット1408が入力されると、入力されたパケットをそのまま出力する。前記パケットは、図14における、パケット1455からパケット1458にあたる。

#### 【0116】

次に、パケット1409が入力されると、これはセクションの先頭パケットであるため、ステップS1303へ遷移する。パケット1409は、差替対象モジュールを構成するセクションの先頭パケットであるため、ステップS1303において、差替元パケット判別手段801はyesと判断し、ステップS1305へ遷移する。入力されるパケットが、パケット1409からパケット1412である場合の処理は、入力パケットがパケット1401からパケット1404である場合の処理と同様の動作をする。

#### 【0117】

パケット1413が入力された時には、ステップS1303でyesと判定しステップS1305へ遷移する。in\_pnum=4であり、この時点での出力対象セクションは、セクション1130であり、out\_pnum=3であるため、in\_pnum>=out\_pnumを満たすため、ステップS1305はyesとなりステップS1306へ遷移する。

#### 【0118】

ステップS1306において、入力されるパケット1413からパケット1415を、差替対象モジュールのセクション1130のパケット1131から1133へ差替えて出力し、ステップS1308に遷移する。ステップS1308において、次に入力されるパケットは、パケット1416であり、これはセクション先頭のパケットではないため、このパケットはNULLパケットに差替えて出力する。前記パケットは、図14におけるパケット1466にあたる。

#### 【0119】

次に入力されるパケット1417はセクション先頭のパケットであるため、ステップS1303に遷移し、パケット1405が入力された時と同様の処理を実行し、パケット1418からパケット1421が入力されるとパケット1401からパケット1404が入力された時と同様の処理を実行する。

#### 【0120】

次に、パケット1422が入力され、ステップS1303でyesと判定され、ステップS1305へ遷移する。パケット1422は2つのパケットから構成されるセクションの先頭であるためin\_pnum=2である。また、この時点



での出力対象セクションは、セクション 1120 であるため `out_pnum=4` である。従って、`in_pnum ≥ out_pnum` の条件を満たさず、ステップ S1305 では `no` と判定され、ステップ S1308 へ遷移する。

#### 【0121】

ステップ S1308 では、次にセクション先頭パケットが入力されるまでは、入力パケットを `NULL` パケットに差替えて出力する。従って、パケット 1422、パケット 1423 を `NULL` に差替えて出力する。これは、図 14 における、パケット 1472、パケット 1473 に相当する。

#### 【0122】

次に入力されるパケット 1424 はセクション先頭のパケットであるため、ステップ S1303 へ遷移し、以後同様にフローチャートに従って処理を実行する。

#### 【0123】

上記処理で、図 14 における入力データ 1400 が入力された時に、図 13 に示す処理フローチャートに基づいて、データ差替装置 80 で出力データ 1450 を出力することが可能となる。

#### 【0124】

以上に示すように、本実施形態によれば、差替対象モジュールのサイズが差替後のモジュールのサイズよりも小さな場合でも、差替対象モジュールを構成するブロックの数が 1 でない限り、差替えないモジュールは入力と全く同様に出力し、かつ、差替対象モジュールを別のモジュールに差替えることが可能となる。

#### 【0125】

##### (実施の形態 4)

本実施の形態は、データを差替える期間が有限である場合に、差替対象モジュールのパケットを差替後のモジュールパケットに差替えて送出した結果、入力された差替対象モジュールのパケット数よりも、差替後モジュールのパケットを余計に送出した状態で差替期間が終了した後に、差替えないモジュールのパケットの遅延が継続しない点で、実施の形態 1 と異なる。以下に、本実施の形態について、実施の形態 1 と異なる点について説明する。

## 【0126】

本実施の形態では、実施の形態1と異なり、差替対象モジュール以外の受信パケットを受信すると一旦入力パケット保持手段104に格納する。通常時には、あるパケットを受信し、入力パケット保持手段104に格納すると、その後パケットをn個受信してから前記パケットを出力する。差替対象モジュールを構成するパケットが入力される毎に、それ以降に入力された差替対象モジュール以外のパケットは、出力タイミングが1パケット分ずつ早くなり、差替後パケット保持手段103からパケットを出力するごとに出力タイミングが1パケット分ずつ遅くなる。

## 【0127】

図15は、本実施形態におけるデータ差替装置10の処理を表すフローチャートである。データ差替装置10は、差替処理を開始すると、ステップS1501において差替対象モジュールの先頭のセクションを差替対象セクションと設定し、差替パケット入力カウンタを0にセットし、ステップS1502へ遷移する。

## 【0128】

ステップS1502においてパケットが入力されると、入力データ受信手段100は入力されたパケットを取得し、差替元パケット判別手段101、モジュール差替手段102へ取得したパケットを通知し、ステップS1503へ遷移する。

## 【0129】

ステップS1503において、差替元パケット判別手段101が、入力データ受信手段100から通知されたパケットが差替対象モジュールを構成するパケットであるかどうかを判定する。入力パケットが差替対象モジュールを構成するパケットであった場合、ステップS1505に遷移し差替パケット入力カウンタに1を加算する。入力パケットが差替対象モジュールを構成するパケットでなかった場合、ステップS1504に遷移し入力パケットを入力パケット保持手段104に格納する。

## 【0130】

ステップS1504、および、ステップS1505の処理終了後には、ステッ

プ S 1 5 0 6 へ遷移する。ステップ S 1 5 0 6 において、データ差替装置 1 0 は、差替パケット入力カウンタの値と差替対象セクションのパケット数を比較し、差替パケット入力カウンタが差替対象セクションのパケット数以上である場合ステップ S 1 5 0 7 へ遷移し、差替パケット入力カウンタが差替対象セクションのパケット数未満である場合ステップ S 1 5 0 9 へ遷移する。

#### 【0131】

ステップ S 1 5 0 7 において、モジュール差替手段 1 0 2 は、入力パケット保持手段 1 0 4 に差替対象セクションを挿入する。具体的なセクション挿入処理は以下に示す通りである。まず、モジュール差替手段 1 0 2 が、入力パケット保持手段 1 0 4 に既に格納されているパケットを先頭から順に走査し、最初に出現するセクションの最終パケットを探す。前記パケットが見つかったら、前記パケットとその次のパケット（次のセクションの先頭パケット）との間に、差替対象セクションを構成するパケットを順番に挿入する。セクションの最終パケットの次にパケットが格納されていない場合、前記パケットの次に差替対象セクションを格納する。また、入力パケット保持手段 1 0 4 にパケットが何も格納されていない場合、入力パケット保持手段 1 0 4 の先頭に差替対象セクションを格納する。

#### 【0132】

ステップ S 1 5 0 7 の処理が終了すると、ステップ S 1 5 0 8 へ遷移し、差替パケット入力カウンタの値から、挿入した差替対象セクションのパケット数を引く。続いて、差替対象セクションの次のセクションを差替対象モジュールから選択し、それを差替対象セクションと設定しなおし、ステップ S 1 5 0 9 へ遷移する。

#### 【0133】

ステップ S 1 5 0 6 で n o と判定した場合と、ステップ S 1 5 0 8 の処理が終わった時には、ステップ S 1 5 0 9 へ遷移し、入力パケット保持手段 1 0 4 に格納されているパケットを出力する。入力パケット保持手段 1 0 4 は、第 1 実施の形態と同様 F I F O 型のメモリ領域であり、パケットは格納された順番で 1 パケットずつ取り出される。ただし、パケットが挿入されている場合には、挿入されたパケットがそれ以前に格納されたパケットよりも先に出力されることもある。

## 【0134】

ステップS1509での処理を終了すると、ステップS1510へ遷移し、差替処理の終了タイミングになったかどうかを判定する。ステップS1510でnoと判定した場合、ステップS1502へ遷移し、次のパケットが入力されるのを待つ。差替処理の終了タイミングに至っており、ステップS1510でyesと判定した場合、ステップS1511へ遷移し差替パケット入力カウンタの値で示される個数のNULLパケットを入力パケット保持手段104に格納する。

## 【0135】

本装置を起動した直後の差替処理開始時など、前に差替処理を実施しておらず、入力パケット保持手段104に何もパケットが格納されていない場合には、初期処理として入力パケット保持手段104にn個のNULLパケットを格納するものとする。

## 【0136】

以下に、図16に示す入力データ1600がデータ差替装置10に入力された場合に、module\_id=0のモジュールを、3パケットからなるセクション1つで構成されるモジュールに差替えて、出力データ1650を出力する処理を例に説明する。ただし、この例においては、nの値を4としている。

## 【0137】

データ差替装置10は、差替処理開始後、ステップS1501において差替対象モジュールの先頭のセクションを差替対象セクションとし、差替パケット入力カウンタを0にセットしステップS1502に遷移する。図16におけるパケット1601が入力されると、ステップS1502において、入力データ受信手段100が前記パケットを取得し、ステップS1503へ遷移する。

## 【0138】

ステップS1503において、差替元パケット判別手段101は、入力パケット1601が差替対象モジュールを構成するパケットであると判定して、ステップS1505へ遷移する。ステップS1505において、データ差替装置10は差替パケット入力カウンタに1を加算し、差替パケット入力カウンタが1とし、ステップS1506へ遷移する。

## 【0139】

差替パケット入力カウンタは1であり、差替対象セクションのパケット数は3であるため、ステップS1506での判定はnoとなり、ステップS1509へ遷移する。ステップS1509において、入力パケット保持手段104に格納されているパケットを出力するが、この時点では、入力パケット保持手段104に、前の差替処理で実施した際に入力パケット保持手段104に格納したパケットが格納されているため、出力パケット1651として、前記既に格納されているパケット、もしくは、前に差替処理を一度もしていない場合には、NULLパケットが出力する。ステップS1509にてパケットを出力した後、ステップS1510へ遷移する。

## 【0140】

ステップS1510では、差替終了タイミングであるかどうかを判定する。本例では、差替終了タイミングは、図16におけるパケット1613が入力された直後とする。パケット1601が入力された時点では、差替終了タイミングに至っていないためnoと判定し、ステップS1502へ遷移する。

## 【0141】

ステップS1502において、次に入力されるパケット1602を受信した場合の処理は、パケット1601を受信した場合と同様の処理をし、再びステップS1502へ遷移する。この時点で、差替パケット入力カウンタの値は2となっている。

## 【0142】

次に、入力パケット1603を受信すると、ステップS1503において、入力パケットは差替対象モジュールを構成するパケットでないためnoと判定し、ステップS1504へ遷移する。ステップS1504では、入力されたパケット1603を入力パケット保持手段104に格納し、ステップS1506へ遷移する。

## 【0143】

ステップS1506以降は、パケット1601が入力された場合と同様の処理をし、ステップS1502へ遷移する。この時点で差替パケット入力カウンタの

値は2のままである。

【0 1 4 4】

ステップS 1 5 0 2において、次に入力されるパケット1 6 0 4からパケット1 6 0 7を受信した場合の処理は、パケット1 6 0 3を受信した場合と同様の処理をし、再びステップS 1 5 0 2へ遷移する。この時点で、差替パケット入力カウンタの値は2のままである。パケット1 6 0 5を受信した時の、ステップS 1 5 0 9で出力処理をする際には、前の差替処理で格納したパケット、もしくは、初期処理で格納したN U L Lパケットを出力パケット1 6 5 1からパケット1 6 5 4で出力し終えているため、入力パケット保持手段1 0 4に格納されている、入力パケット1 6 0 3が出力される（パケット1 6 5 5）。以降の出力は、入力パケット1 6 0 3以降に入力されたパケットとなる。

【0 1 4 5】

次に、パケット1 6 0 8が入力されると、ステップS 1 5 0 3において、y e sの判定となり、ステップS 1 5 0 5に遷移して、差替パケット入力カウンタに1を加算する。この結果、差替パケット入力カウンタが3となり、ステップS 1 5 0 6においてy e sと判定し、ステップS 1 5 0 7へ遷移する。

【0 1 4 6】

ステップS 1 5 0 7において、入力パケット保持手段1 0 4に差替対象セクションを挿入する処理を実施するが、この時点で入力パケット保持手段1 0 4に格納されているパケットは、パケット1 6 0 6と入力パケット1 6 0 7の2つのパケットである。パケット1 6 0 6は直前のパケットと同一セクションであるため、差替対象セクションを挿入することは出来ないが、パケット1 6 0 7は直前のパケットとは別のセクション先頭パケットであるため、パケット1 6 0 7の前に差替対象セクションを挿入し、ステップS 1 5 0 8に遷移する。

【0 1 4 7】

ステップS 1 5 0 8においては、差替対象セクションの次のセクションを差替対象セクションとするが、本例では差替対象モジュールが1つのセクションからなるモジュールであるため、差替対象セクションは変化しない。ステップS 1 5 0 8においては、差替パケット入力カウンタを0にセットする処理も実施する。

## 【0148】

この時点で、差替パケット入力カウンタは0で、入力パケット保持手段104には、パケット1606、差替対象セクションを構成するパケット3つ、パケット1607の合計5つのパケットが前記順番で格納されている。

## 【0149】

その後、ステップS1509において、入力パケット保持手段104に格納されているパケット1606を出力し（出力パケット1658）、ステップS1502へ遷移する。

## 【0150】

次に、パケット1609が入力されると、パケット1601が入力された時と同様の処理を実施する。この時にステップS1509では入力パケット保持手段104の先頭には差替対象セクションの先頭のパケットが格納されているため、差替対象セクションの先頭のパケットを出力する。パケット1609の処理が終了した時点で、差替パケット入力カウンタの値は1になる。

## 【0151】

パケット1610からパケット1613が入力された時には、パケット1603と同様の処理を実施する。ただし、パケット1603が入力された時に差替終了タイミングになるため、ステップS1510での判定がyesとなり、ステップS1511へ遷移する。ステップS1511へ遷移する。ステップS1511では、その時点での差替パケット入力カウンタが1であるため、NULLパケット1つを入力パケット保持手段104に格納する。前記処理の結果、入力パケット保持手段104には、パケット1611、パケット1612、パケット1613、NULLパケットの4つのパケットが格納されている状態となる。ステップS1511では、差替パケット入力カウンタの値を0にセットする処理も実施する。

## 【0152】

ステップS1511の処理を完了すると、差替処理が終了するため、以降の入力パケットは、一旦必ず入力パケット保持手段104に格納され、出力パケットは入力パケット保持手段104に格納されているパケットを順番に出力する。

## 【0153】

上記処理で、図16における入力データ1600が入力された時に、図15に示す処理フローチャートに基づいて、データ差替装置10で出力データ1650を出力することが可能である。

## 【0154】

なお、図15のステップS1506において、差替パケット入力カウン트의値が差替対象セクションのパケット数以上であるという判定を別の判定に変更し、差替えるパケットの平均遅延を制御することも可能である。例えば、前記判定条件を、差替パケット入力カウン트의値が差替対象セクションのパケット数の半分以上であると変更することで、差替対象モジュールを構成するモジュールの遅延パケット数 $-n$ の絶対値の平均を最小にすることも可能である。

## 【0155】

(実施の形態5)

本実施の形態は、複数個のモジュールを差替対象モジュールとし、差替える対象のモジュール1つに対して、差替後のモジュール1つが対応することを特徴とする。差替超過カウントや差替後のモジュール実体を、差替対象とするモジュール毎に持つ点で実施の形態1と異なる。以下に、本実施の形態について、実施の形態1と異なる点について説明する。

## 【0156】

図17は、本実施形態におけるデータ差替装置10の処理を表すフローチャートである。データ差替装置10は、差替処理を開始すると、ステップS1700において全ての差替超過カウントを0にセットする。本実施の形態では、データ差替装置10は差替超過カウントと、差替後のモジュールを差替える対象とするモジュール数分だけ持つ。例えば、モジュール $n$ 個を差替える場合、入力モジュールの`module Id`とそれに対応する差替超過カウントをそれぞれ $n$ 個持つ。入力ストリーム中に含まれる差替える対象となるモジュールを差替対象モジュールと呼び、 $i$ 番目の差替対象モジュールを差替対象モジュール $[i]$ と呼ぶ。差替対象モジュールを差替えることで、出力する差替後のモジュールのことを差替後モジュールと呼び、 $i$ 番目の差替後モジュールを差替後モジュール $[i]$ と



呼ぶ。また、 $i$  番目の差替対象モジュールに対応する差替超過カウントを差替超過カウント  $[i]$  と呼ぶ。ステップ S1700 においては、差替超過カウント  $[0]$  から差替超過カウント  $[n]$  の全ての差替超過カウントを 0 にセットする処理を実施し、ステップ S1701 に遷移する。

#### 【0157】

ステップ S1701 においてパケットが入力されると、入力データ受信手段 100 は入力されたパケットを取得し、差替元パケット判別手段 101、モジュール差替手段 102 へ取得したパケットを通知し、ステップ S1702 へ遷移する。

#### 【0158】

ステップ S1702 においては、差替元パケット判別手段 101 が、入力パケットが差替対象モジュールを構成するパケットであるかどうかを判定し、入力パケットが差替対象モジュールを構成するパケットでない場合、ステップ S1703 へ遷移し、入力パケット入力パケット保持手段 104 に格納する処理を実施する。

#### 【0159】

ステップ S1702 で入力パケットが差替対象モジュールを構成するパケットでないと判定した場合や、ステップ S1703 の処理が終了するとステップ S1704 に遷移し、直前にセクション途中のパケットを出力したかどうかの判定をする。直前にセクションの最後以外にあたるパケットを出力していた場合にはステップ S1705 に遷移する。差替処理を開始した直後で直前に何もパケットを出力していない場合や、直前にセクションの最後にあたるパケットを出力していた場合には、ステップ S1711 に遷移する。

#### 【0160】

ステップ S1704 において、直前にセクション途中のパケットを出力したと判定した場合、ステップ S1705 において、直前に出力したパケットの次のパケットを出力する。この処理は実施の形態 1 におけるステップ S603 と同様である。ステップ S1705 の処理実行後、ステップ S1706 へ遷移し、ステップ S1705 で出力したパケットが入力パケット保持手段 104 に格納されてい

たパケットであるかどうかを判定する。ステップS1705で出力したパケットが入力パケット保持手段104に格納されていたパケットである場合、ステップS1707に遷移し、そうでない場合ステップS1709に遷移する。

#### 【0161】

ステップS1707において、入力パケットが差替対象モジュールのいずれかのパケットである場合には、差替超過カウント[i]から1を減算する(ステップS1708)。ただしiは、入力パケットは、差替対象モジュールのうちi番目のモジュールを構成するパケットであるとする。一方ステップS1709においても、ステップS1707と同様の判定をしnoと判定した場合、差替超過カウント[i]に1を加算する(ステップS1710)。上記処理を完了すると、ステップS1701へ遷移する。

#### 【0162】

ステップS1704においてnoの判定をした場合、ステップS1711へ遷移し、ステップS1707、ステップS1709と同様の判定をする。ステップS1711でyesの判定をした場合、差替超過カウント[i]から1を減算する(ステップS1712)。ステップS1713において、全てのモジュールの差替超過カウントを比較し、差替超過カウントが最小のモジュールの位置を検出し、そのモジュールの順番をjとし、ステップS1714に遷移する。

#### 【0163】

ステップS1714において、差替超過カウント[j]の値が0未満であるかどうかを判定する。差替超過カウント[j]が0以上である場合、ステップS1717へ遷移し、入力パケット保持手段104に格納されているパケットを出力しステップS1701へ遷移する。差替超過カウント[j]が0未満である場合、ステップS1715へ遷移し、差替超過カウント[j]に1を加算しステップS1716へ遷移する。ステップS1716では、差替後パケット保持手段103に格納されているj番目のモジュールのうち、次に送出するセクションの先頭パケットを出力してステップS1701に遷移する。ここで、次に送出するセクションは、実施の形態1の処理フローチャート図6におけるステップS612で選択するセクションと同様の条件で決定する。

## 【0164】

以上の処理で、複数のモジュールを差替えることが可能である。

## 【0165】

なお、差替えるモジュールに優先度を付けることも可能である。その場合、差替えるモジュールに優先度を付けておき、ステップS1713において、差替超過カウントが0未満で優先度の最も高いモジュールをjと設定することで、優先度の高い差替モジュールの遅延を最小限にして出力することが可能となる。もちろん、全てのモジュールに対して優先順位を一意にする必要は無く、優先度の同じモジュールが存在してもよい。その場合、優先度の同じモジュール同士は、差替超過カウントで比較してjを決定する。差替超過カウントが同じ場合は、順番の若い方などどちらか適当な方を選択すれば良い。

## 【0166】

なお、差替えないモジュールにも優先度を付けることも可能である。その場合、ステップS1714において、差替超過カウントが0未満の差替モジュール全ての優先度が、差替えないモジュールの優先度よりも低い場合、差替えないモジュールを優先的に出力（入力パケット保持手段104のパケットを優先的に出力）することで実現可能である。

## 【0167】

なお、差替対象モジュール全体の合計の差替超過カウントを持つことも可能である。具体的には、図17に示す処理フローチャートにおいて、ステップS1712、および、ステップS1708で差替超過カウント[i]から1を減算する時に同時に、差替対象モジュール全体の差替超過カウントからも1を減算し、ステップS1715、および、ステップS1710で差替超過カウント[i]に1を加算する時に同時に、差替対象モジュール全体の差替超過カウントにも1を加算する処理を追加し、ステップS1714の判定時に全体の差替超過カウントが0未満であるという条件も追加することで、差替対象モジュール全体の合計の差替超過カウントを用いた処理になる。差替対象モジュール全体の合計の差替超過カウントを用いることで、差替対象モジュール数に比例して、差替えないモジュールの最大遅延が大きくなることを防ぐことが可能となる。

## 【0 1 6 8】

なお、本発明では、入力されたモジュールは、入力パケット保持手段にパケット単位で格納しパケット単位で出力するが、図 5 に示すように入力モジュールも差替後パケット保持手段にモジュール単位で格納し、入力パケット 1 個に対して、対応するパケット 1 個を差替後パケット保持手段から出力するようにしてもよい。

## 【0 1 6 9】

なお、本発明の実施の形態では M P E G 2 規格に従ってデータが伝送されることを前提として記述しているが、もちろん、データをパケット化し繰り返し送出する伝送方式のストリームであれば、M P E G 2 規格に従っていなくとも本発明が有効であることは言うまでもない。

## 【0 1 7 0】

## 【発明の効果】

以上、詳述したように、本発明によれば、繰り返し入力されるデータの一部を、帯域を保ったまま差替え、かつ、差替えないデータの遅延をデータの 1 ブロックの最低単位に抑えることが可能となる。本発明によって、発局の制作したデータコンテンツを受信した受局が、帯域に影響を与えることなく、入力の一部のデータを受局向けに差替えて出力することで、CM 差替等の運用が可能となる。

## 【図面の簡単な説明】

## 【図 1】

本発明の装置の基本的な構成図

## 【図 2】

本発明の装置への入力となるデータストリームを示す図

## 【図 3】

M P E G 2 規格における、モジュール、セクション、パケットの関係を示す図

## 【図 4】

本発明の実施の形態 1 における入出力ストリームの例を示す図

## 【図 5】

差替後パケット保持手段に格納されるモジュール構成の例を示す図

**【図 6】**

本発明の実施の形態 1 におけるパケット出力の処理フローチャート

**【図 7】**

本発明の実施の形態 1 で示す差替対象モジュールの例を示す図

**【図 8】**

本発明の実施の形態 2 における本発明の装置の構成図

**【図 9】**

blockSize=700 の場合の、サイズが 3000 オクテットのモジュール構造を示す図

**【図 10】**

本発明の実施の形態 2 における、3000 オクテットのモジュールを 2000 オクテットのモジュールに差替える例を示す図

**【図 11】**

blockSize=700 の場合の、サイズが 2000 オクテットのモジュール構造を示す図

**【図 12】**

本発明の実施の形態 2 におけるパケット出力の処理フローチャート

**【図 13】**

本発明の実施の形態 3 におけるパケット出力の処理フローチャート

**【図 14】**

本発明の実施の形態 3 における、3000 オクテットのモジュールを 2000 オクテットのモジュールに差替える例を示す図

**【図 15】**

本発明の実施の形態 4 におけるパケット出力の処理フローチャート

**【図 16】**

本発明の実施の形態 4 における入出力ストリームの例を示す図

**【図 17】**

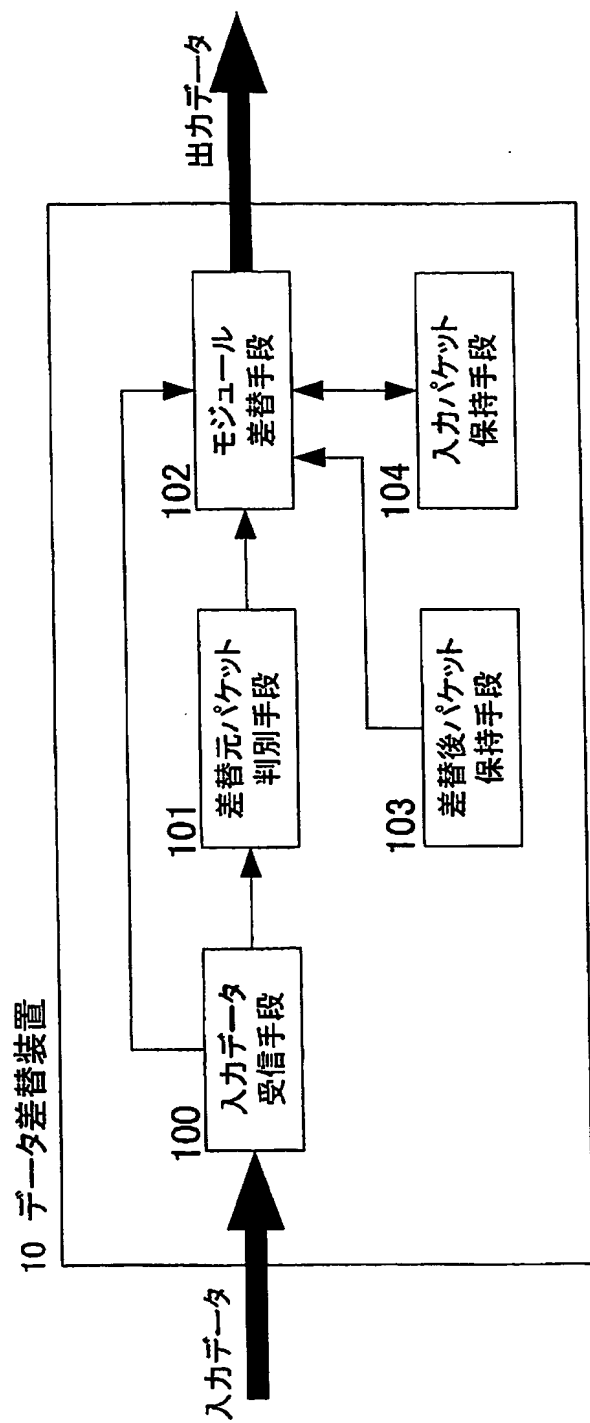
本発明の実施の形態 5 におけるパケット出力の処理フローチャート

**【符号の説明】**

- 1 0 データ差替装置
- 1 0 0 入力データ受信手段
- 1 0 1 差替元パケット判別手段
- 1 0 2 モジュール差替手段
- 1 0 3 差替後パケット保持手段
- 1 0 4 入力パケット保持手段

【書類名】 図面

【図 1】



【図 2】

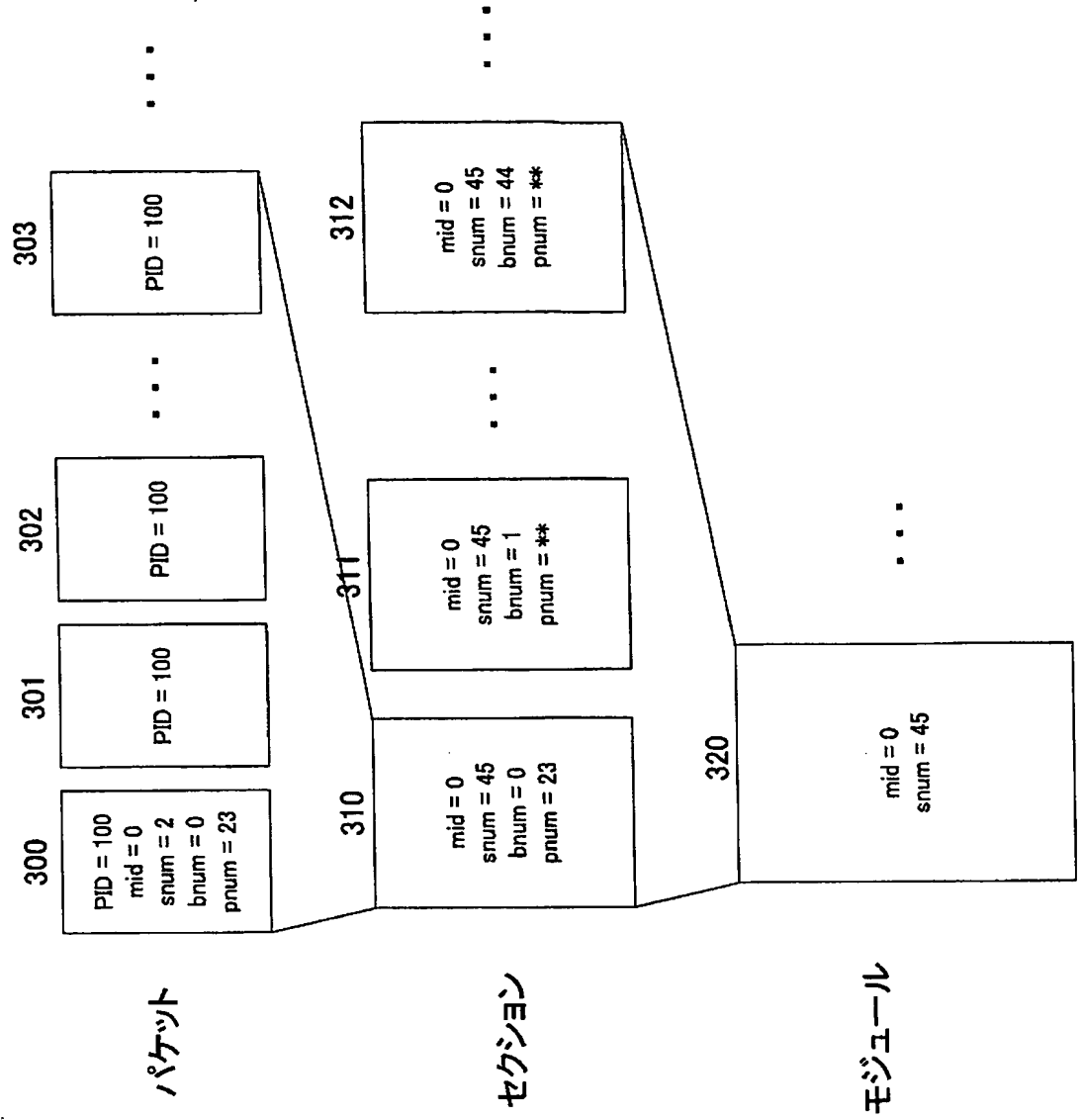
200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216
PID = 100 mid = 0 snum = 2 bnum = 0 pnum = 3	PID = 101 mid = 0 snum = 5 bnum = 0 pnum = 4	PID = 100	PID = 100	PID = 101	PID = 100 mid = 0 snum = 2 bnum = 1 pnum = 4	PID = 101	PID = 100	PID = 100	PID = 101	PID = 100	PID = 101 mid = 0 snum = 5 bnum = 1 pnum = 2	PID = 100 mid = 1 snum = 10 bnum = 0 pnum = 4	PID = 100	PID = 101	PID = 100	PID = 100

↑  
時間

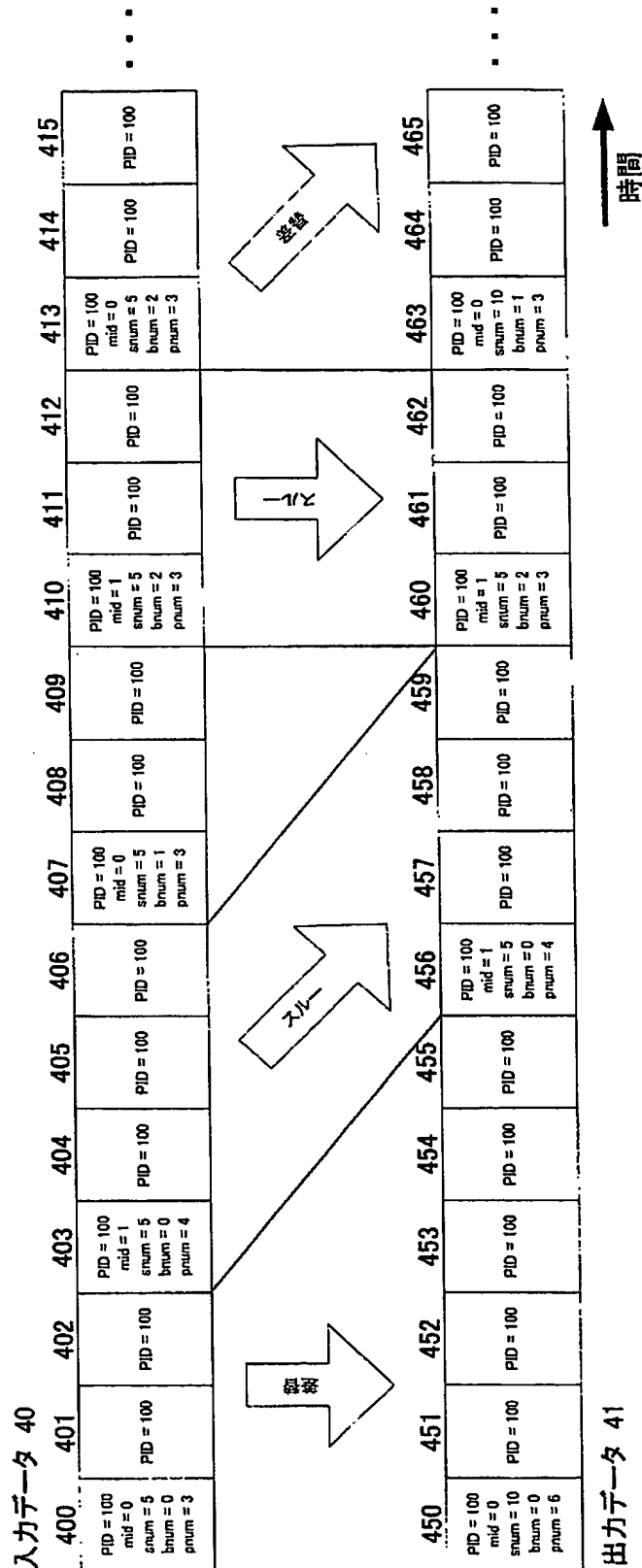
mid : moduleId  
bnum : blockNumber  
snum : そのモジュールが何個のセクションから成るかを表す値  
pnum : そのセクションが何個のパケットから成るかを表す値



【図 3】



【図 4】



【図 5】

## 差替対象モジュール 50

## blockNumber = 0 のセクション

500      501      502      503      504      505

mid = 0 snum = 10 bnum = 0 pnum = 6					
--	--	--	--	--	--

## blockNumber = 1 のセクション

510      511      512

mid = 0 snum = 10 bnum = 1 pnum = 3		
--	--	--

## blockNumber = 2 のセクション

520      521      522      523      524

mid = 0 snum = 10 bnum = 2 pnum = 5				
--	--	--	--	--

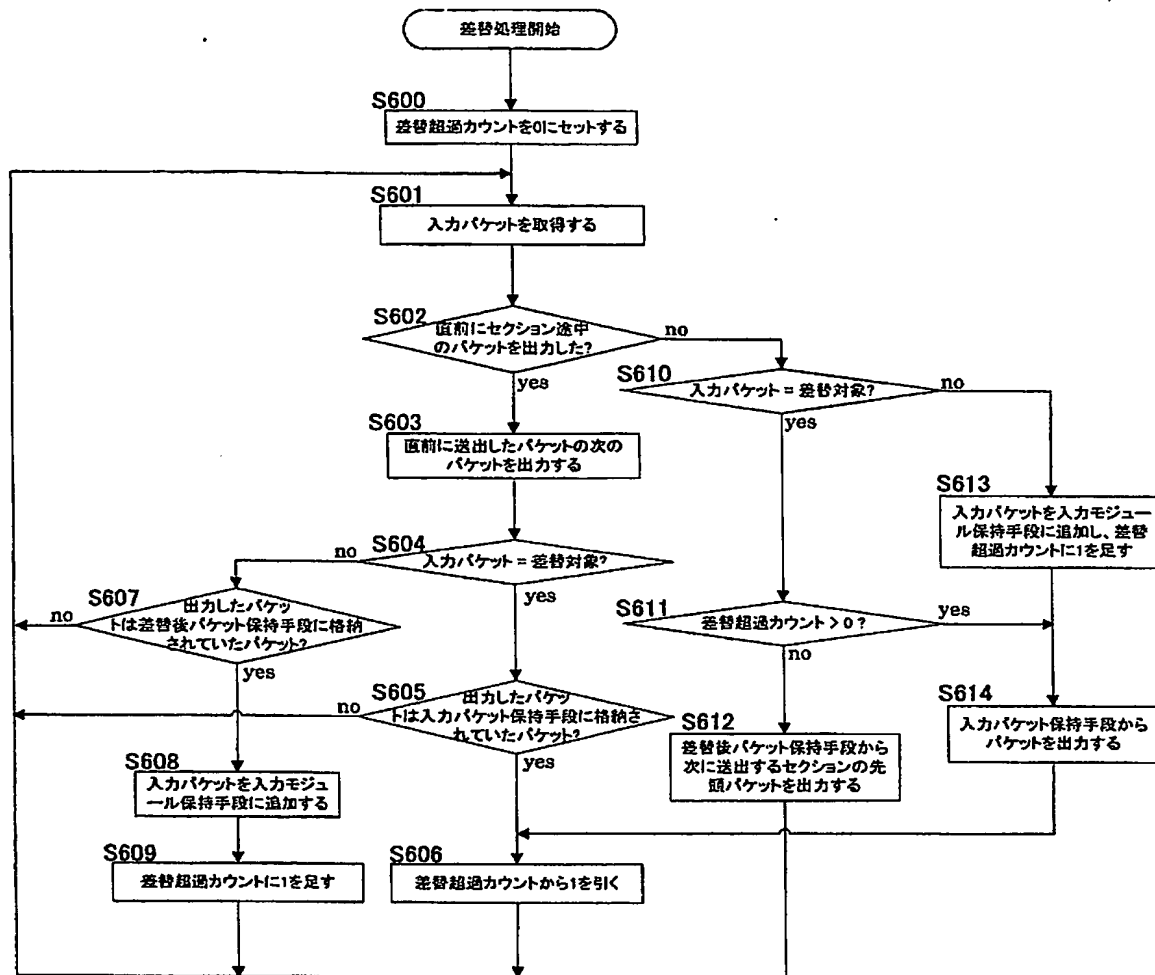
▪  
▪  
▪

## blockNumber = 9 のセクション

530      531      532      533

mid = 0 snum = 10 bnum = 9 pnum = 4			
--	--	--	--

【図 6】



【図 7】

## 差替対象モジュール 70

blockNumber = 0 のセクション

700      701      702      703      704      705

mid = 3 snum = 10 bnum = 0 pnum = 6					
--	--	--	--	--	--

blockNumber = 1 のセクション

710      711      712

mid = 3 snum = 10 bnum = 1 pnum = 3		
--	--	--

blockNumber = 2 のセクション

720      721      722      723      724

mid = 3 snum = 10 bnum = 2 pnum = 5				
--	--	--	--	--

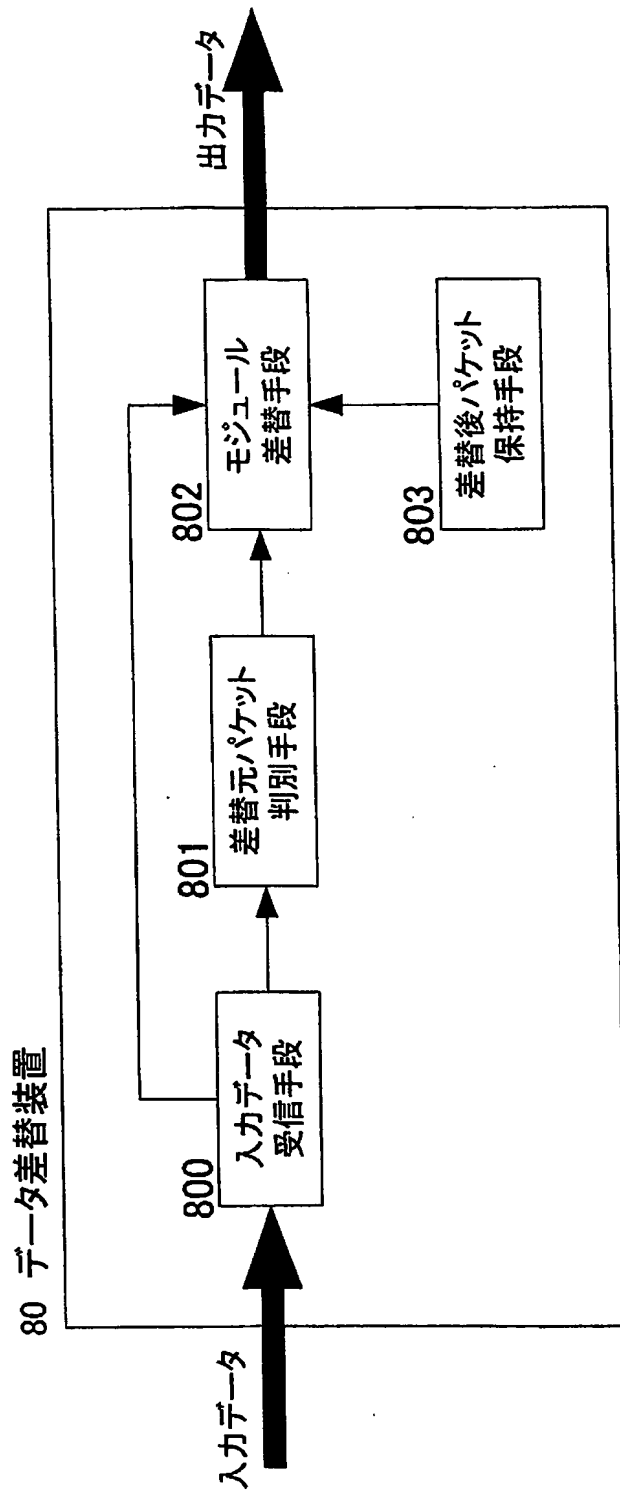
▪  
▪  
▪

blockNumber = 9 のセクション

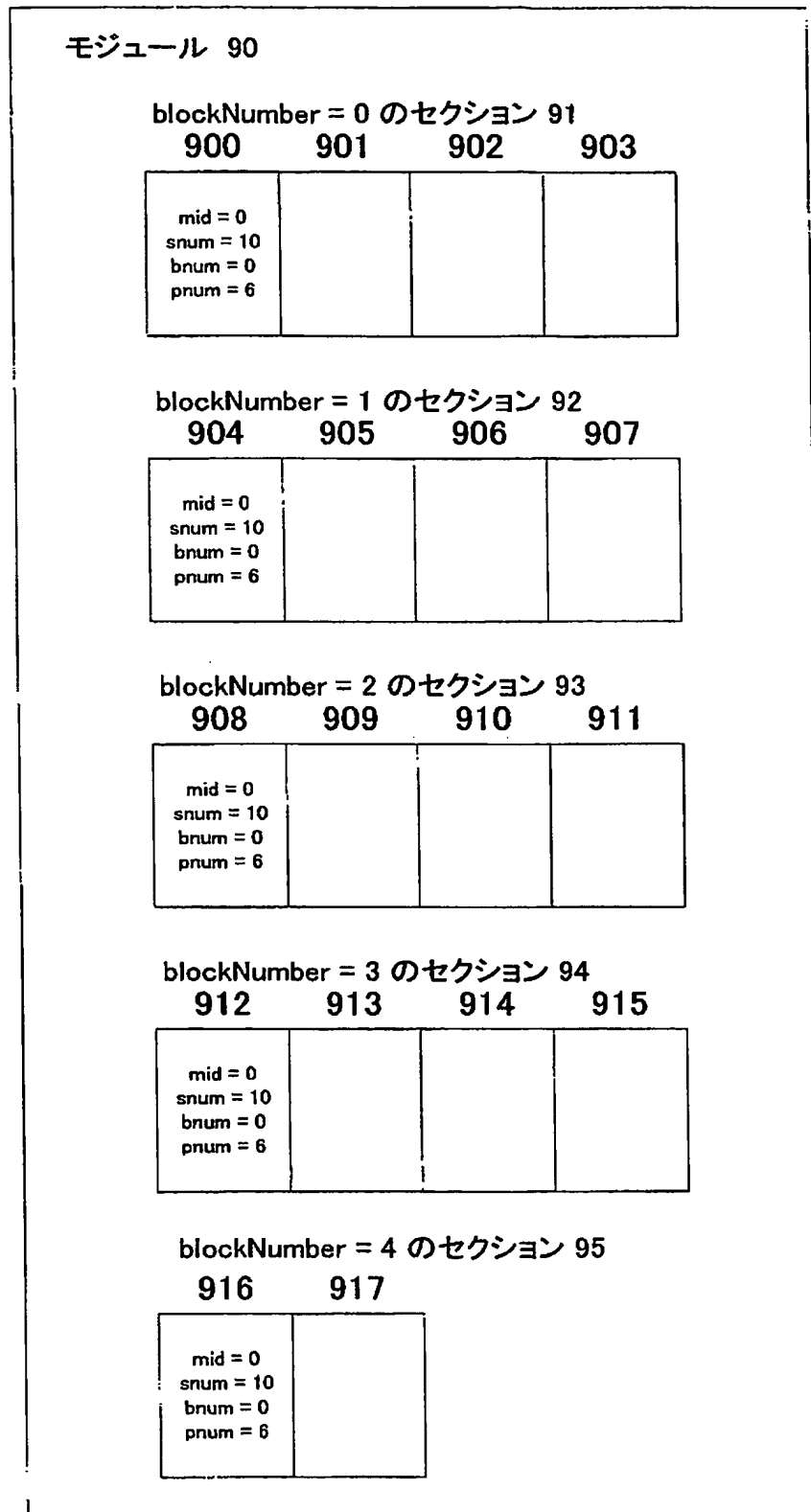
730      731      732      733

mid = 3 snum = 10 bnum = 9 pnum = 4			
--	--	--	--

【図 8】



【図 9】



【図 10】

入力データ 1000

1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012
PID=100 mid = 0 snum = 5 bnum = 0 pnum = 4	PID=100	PID=100	PID=100	PID=100 mid = 1 snum = 2 bnum = 0 pnum = 4	PID=100	PID=100	PID=100	PID=100 mid = 0 snum = 5 bnum = 1 pnum = 4	PID=100	PID=100	PID=100
↓ 追加				↓ スルー				↓ 追加			
1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062
PID=100 mid = 0 snum = 3 bnum = 0 pnum = 4	PID=100	PID=100	PID=100	PID=100 mid = 1 snum = 2 bnum = 0 pnum = 4	PID=100	PID=100	PID=100	PID=100 mid = 0 snum = 3 bnum = 1 pnum = 4	PID=100	PID=100	PID=100

出力データ 1050

入力データ 1000

1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024
PID=100 mid = 0 snum = 5 bnum = 2 pnum = 4	PID=100	PID=100	PID=100	PID=100 mid = 1 snum = 2 bnum = 1 pnum = 1	PID=100 mid = 0 snum = 5 bnum = 3 pnum = 4	PID=100	PID=100	PID=100	PID=100 mid = 0 snum = 5 bnum = 3 pnum = 2	PID=100	PID=100 mid = 1 snum = 2 bnum = 0 pnum = 4
↓ 追加			↓ NULL 置換	↓ スルー	↓ NULL 置換					↓ スルー	
1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073	1074
PID=100 mid = 0 snum = 3 bnum = 2 pnum = 3	PID=100	PID=100	NULL	PID=100 mid = 1 snum = 2 bnum = 1 pnum = 1	NULL	NULL	NULL	NULL	NULL	NULL	PID=100 mid = 1 snum = 2 bnum = 0 pnum = 4

出力データ 1050



【図 11】

## モジュール 1100

blockNumber = 0 のセクション 1110

1111      1112      1113      1114

mid = 0 snum = 10 bnum = 0 pnum = 6			
--	--	--	--

blockNumber = 1 のセクション 1120

1121      1122      1123      1124

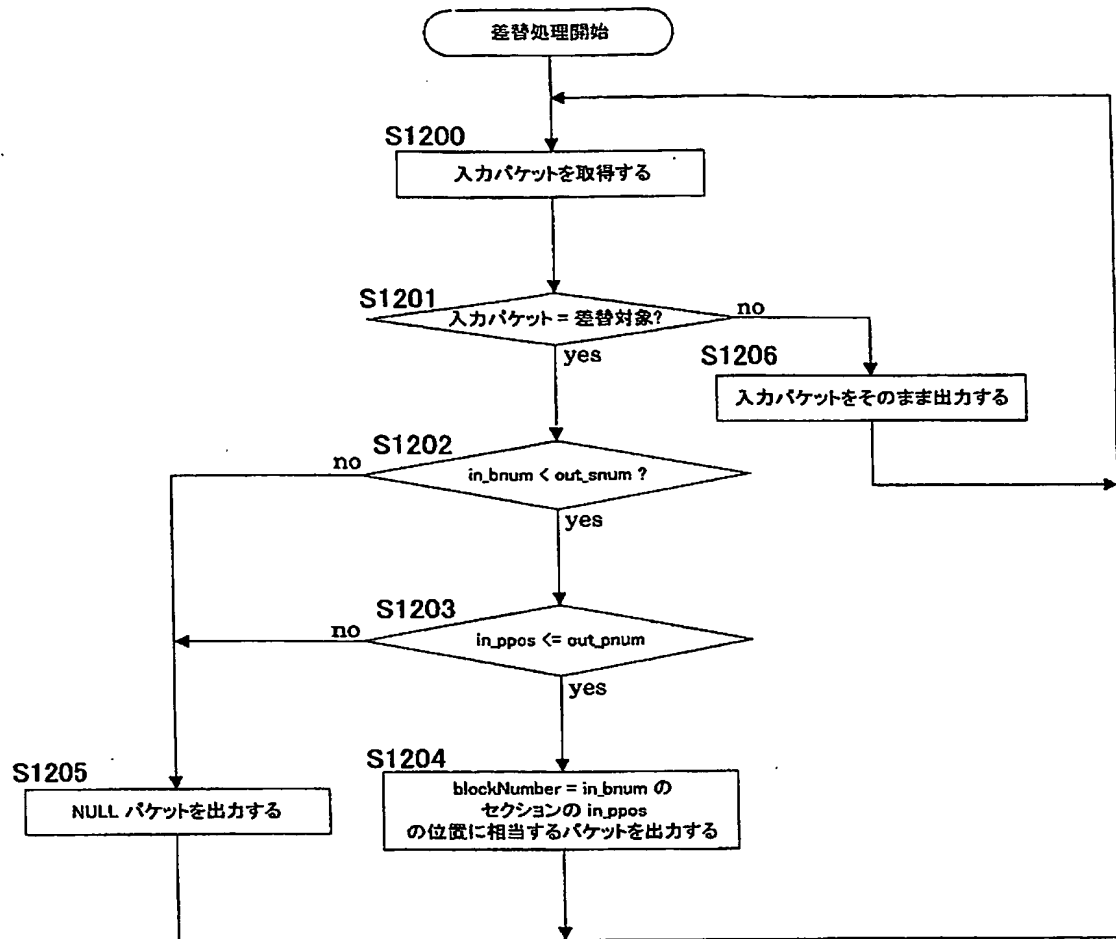
mid = 0 snum = 10 bnum = 0 pnum = 6			
--	--	--	--

blockNumber = 2 のセクション 1130

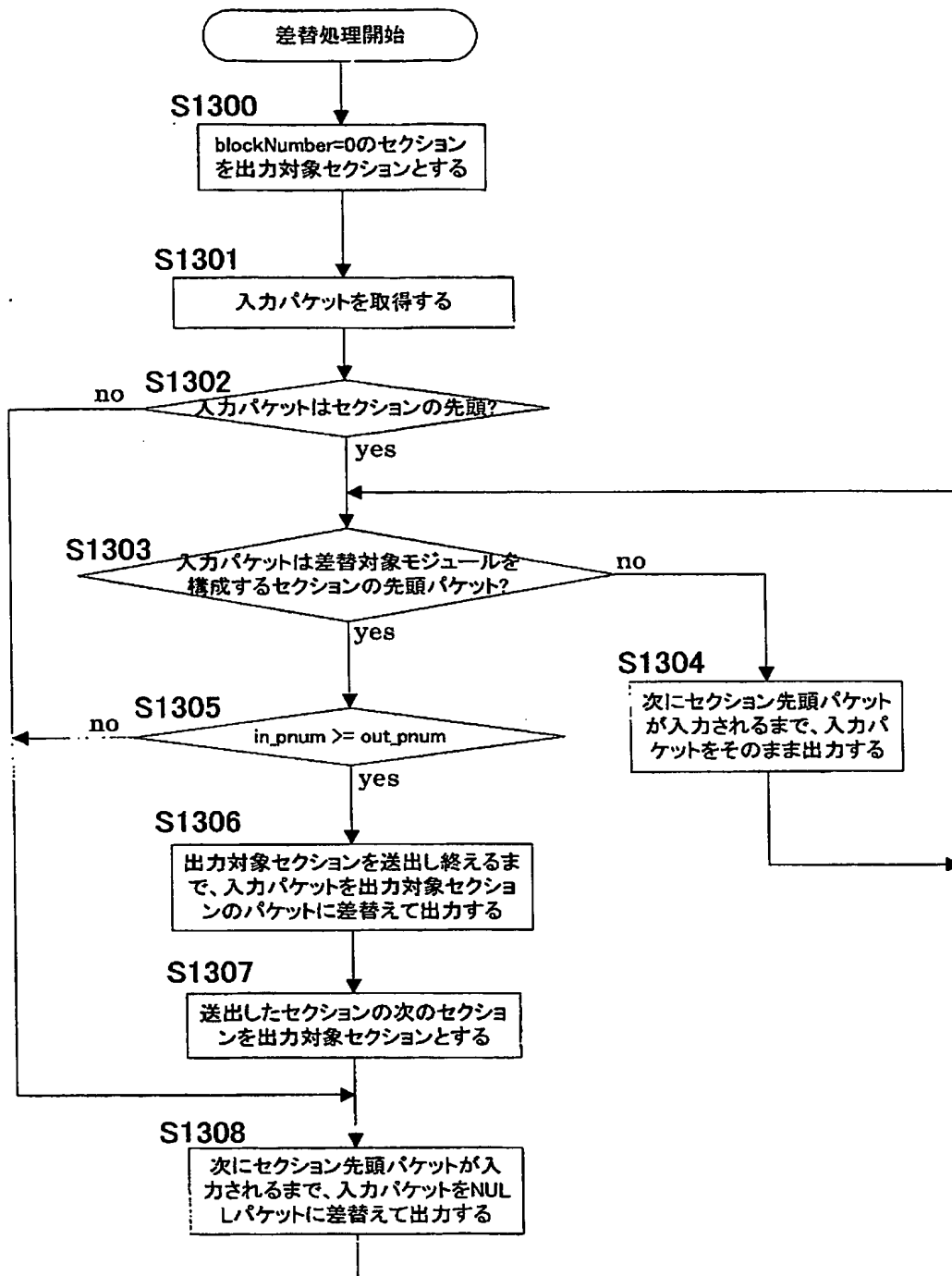
1131      1132      1133

mid = 0 snum = 10 bnum = 0 pnum = 6		
--	--	--

【図 12】



【図 13】



【図 14】

入力データ 1400

1401	1402	1403	1404	1405	1406	1407	1408	1409	1410	1411	1412
PID=100 mid = 0 snum = 5 bnum = 0 pnum = 4	PID=100	PID=100	PID=100	PID=100 mid = 1 snum = 2 bnum = 0 pnum = 4	PID=100	PID=100	PID=100	PID=100 mid = 0 snum = 5 bnum = 1 pnum = 4	PID=100	PID=100	PID=100
↓ 結合				↓ スルー				↓ 結合			
1451	1452	1453	1454	1455	1456	1457	1458	1459	1460	1461	1462
PID=100 mid = 0 snum = 3 bnum = 0 pnum = 4	PID=100	PID=100	PID=100	PID=100 mid = 1 snum = 2 bnum = 0 pnum = 4	PID=100	PID=100	PID=100	PID=100 mid = 0 snum = 3 bnum = 1 pnum = 4	PID=100	PID=100	PID=100

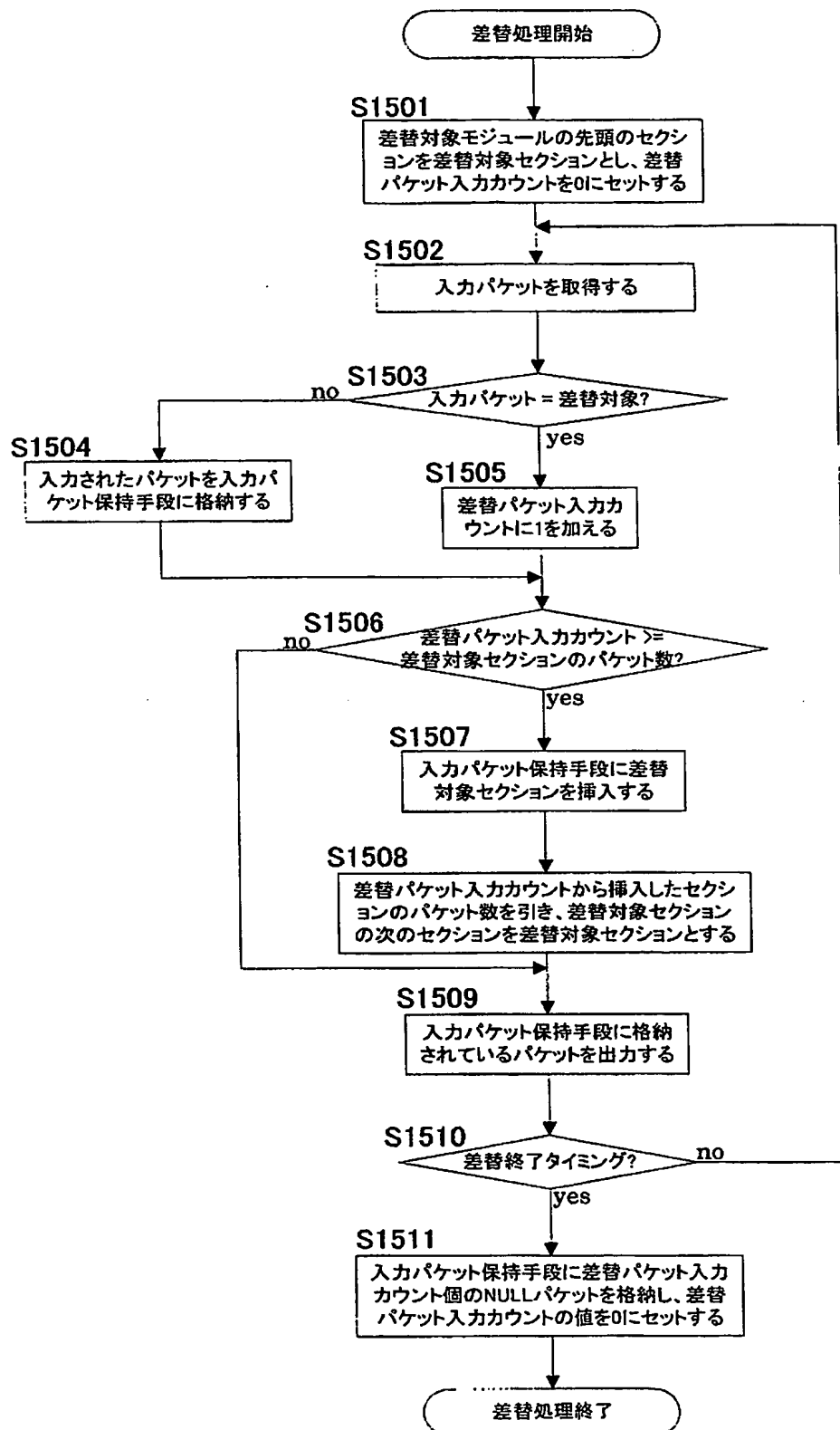
出力データ 1450

入力データ 1400

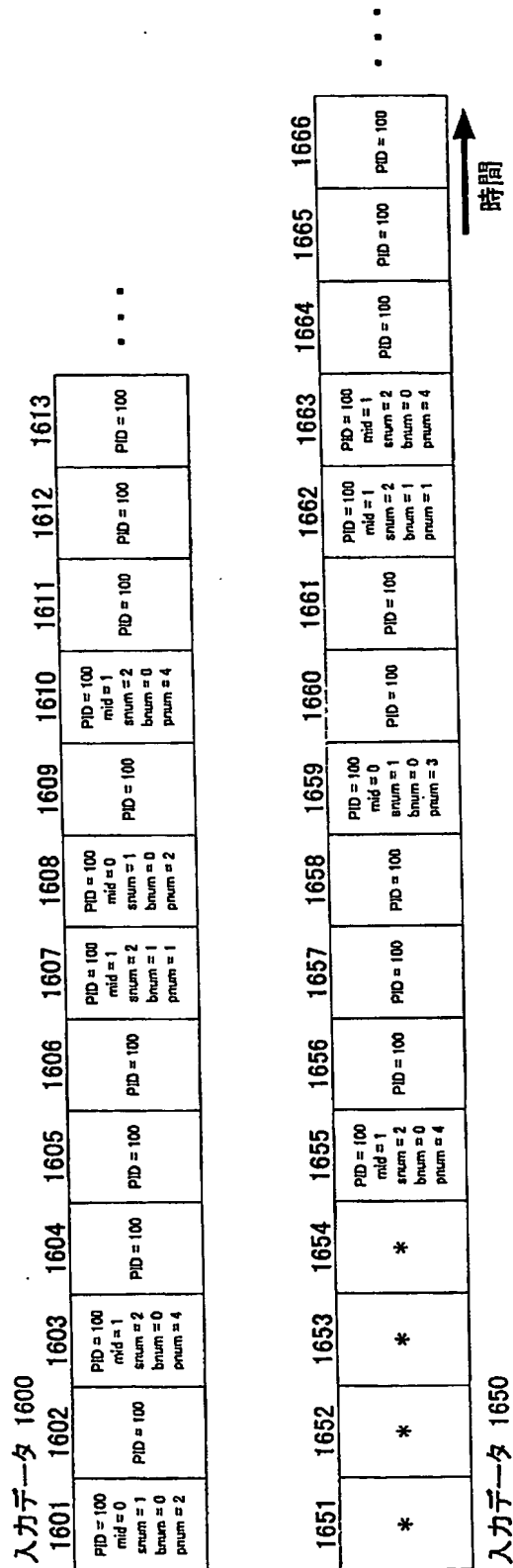
1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423	1424
PID=100 mid = 0 snum = 5 bnum = 2 pnum = 4	PID=100	PID=100	PID=100	PID=100 mid = 1 snum = 2 bnum = 1 pnum = 1	PID=100 mid = 0 snum = 5 bnum = 3 pnum = 4	PID=100	PID=100	PID=100	PID=100 mid = 0 snum = 5 bnum = 3 pnum = 2	PID=100	PID=100 mid = 1 snum = 2 bnum = 0 pnum = 4
↓ 結合			↓ NULL 置換		↓ 結合			↓ NULL 置換		↓ スルー	
1463	1464	1465	1466	1467	1468	1469	1470	1471	1472	1473	1474
PID=100 mid = 0 snum = 3 bnum = 2 pnum = 3	PID=100	PID=100	NULL	PID=100 mid = 1 snum = 2 bnum = 1 pnum = 1	PID=100 mid = 0 snum = 3 bnum = 0 pnum = 4	PID=100	PID=100	PID=100	NULL	NULL	PID=100 mid = 1 snum = 2 bnum = 0 pnum = 4

出力データ 1450

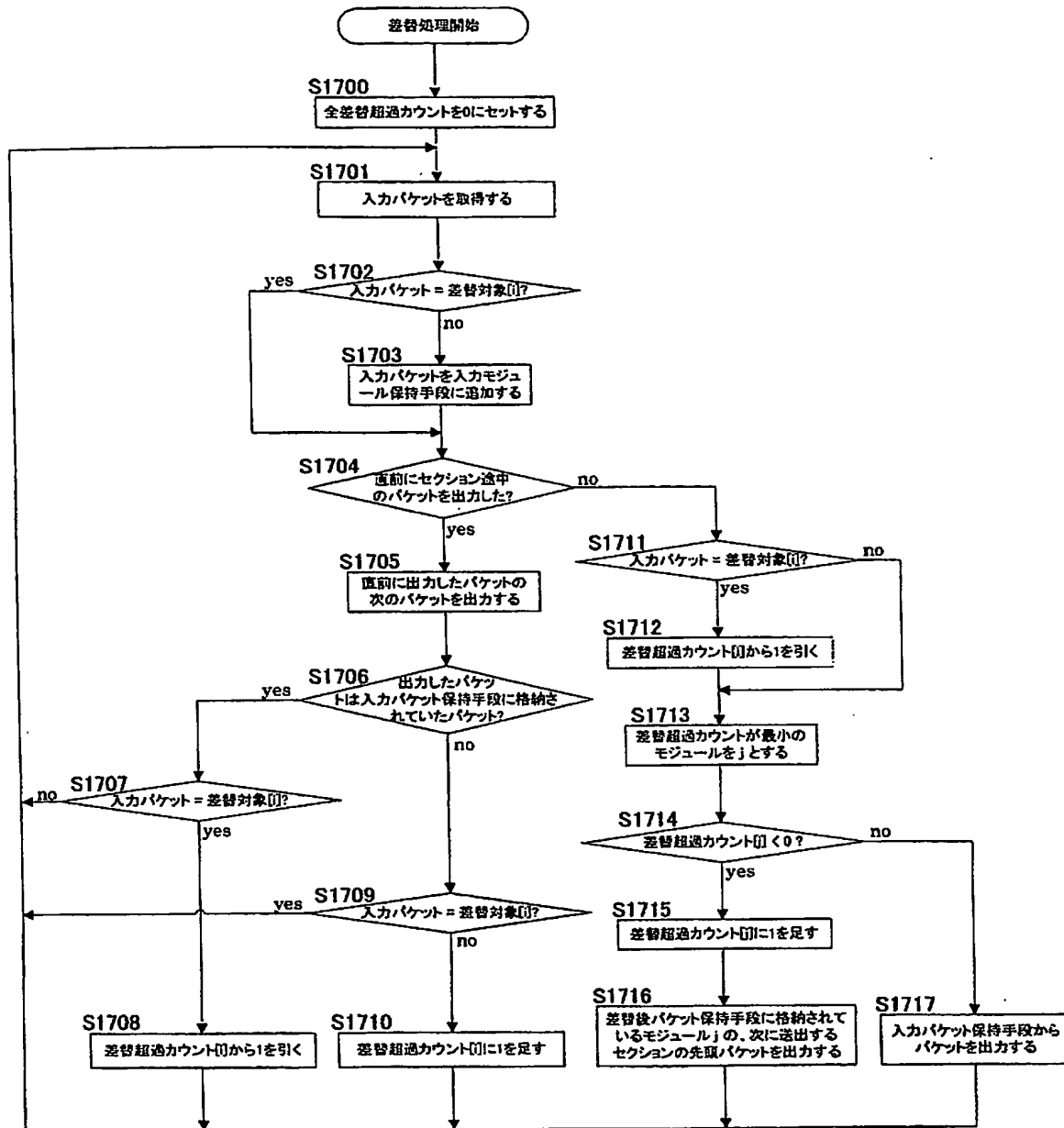
【図15】



【図 16】



【図 17】



【書類名】 要約書

【要約】

【課題】 繰り返し入力されるデータの一部を単純に差替えると、差替えるデータと入力データのサイズが異なる場合、入出力の間でデータにずれが発生する。このことにより、入力よりもサイズの大きなデータに1対1で差替えたストリームを受信した時に、元のストリームと比較して、差替えていないデータの応答速度が遅くなってしまう。

【解決手段】 各データの入力帯域と出力帯域を可能な限り等しくし、差替えるデータと差替えないデータを伝送の最小単位で重畳して出力することで、差替えないデータの遅延を最小にする。

【選択図】 図1



特願2002-274539

出願人履歴情報

識別番号

[000005821]

1. 変更年月日

1990年 8月28日

[変更理由]

新規登録

住 所

大阪府門真市大字門真1006番地

氏 名

松下電器産業株式会社